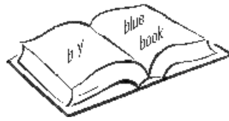
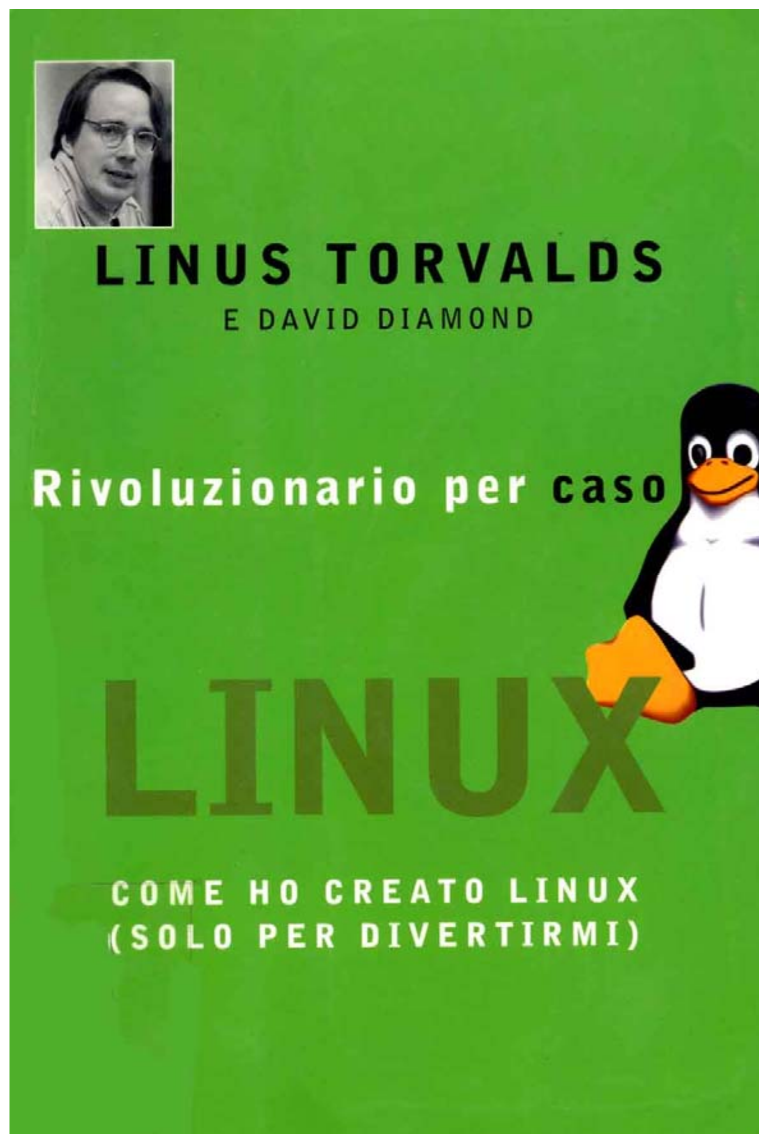
Linus Torvalds e David Diamond

Rivoluzionario per caso

Come ho creato Linux
(solo per divertirmi)





Prima edizione: ottobre 2001
Traduzione dall'Inglese di Fabio Paracchini
Titolo originale dell'opera:
Just for Fun

www.garzantilibri.it

Indice

A Tove e Patricia, Daniela e Celeste. Ho sempre desiderato essere circondato da giovani donne, e voi avete fatto in modo che fosse così.

Per Tia e Kaley. Cavoli se ho avuto fortuna.

Questi non sarebbero dei veri ringraziamenti se non lasciassimo cadere - come a caso - qualche nome importante. E allora:

Ringraziamo il nostro editor, Adrian Zackheim, che ci ha fatto da balia; Eric Richnow, l'assistant editor di HarperCollins che su questo progetto era sempre più aggiornato di noi; i nostri agenti Bill Gladstone della Waterside Productions e Kris Dahl della ICM, che non avrebbero potuto essere più veloci a spedirci i nostri assegni; Sara Torvalds, che ha la migliore memoria della penisola finnoscandinava (e lavora in tre lingue) e William e Ruth Diamond, che hanno letto il manoscritto originale continuando a ripetere:

«No, davvero, è buono. »

Guardandolo crescere non potevo non pormi una domanda: messo com’è, come diavolo farà a incontrare una ragazza carina?

Anna Torvalds

Post-It dalla rivoluzione

C'era un rivoluzione in corso - tra le tante - nell'euforia degli ultimi anni del xx secolo. Nello spazio di una notte, o almeno così pareva, il sistema operativo Linux aveva catalizzato l'attenzione di tutto il mondo. Era esploso dalla cameretta del suo creatore, Linus Torvalds, fino a generare una specie di culto religioso da parte di un esercito di geek[1] *semimilitanti. All'improvviso si slava infiltrando nelle roccaforti aziendali che controllavano il pianeta, contava milioni di utenti in tutti i continenti (Antartide compresa) e persino nello spazio, se vogliamo considerare le basi orbitanti della NASA. Questo sistema operativo gestiva la maggior parte dei server che diffondevano nel mondo il contenuto del World Wide Web e il suo stesso modello di sviluppo, un'intricata rete di centinaia, di migliaia di programmatori volontari, era cresciuto fino a diventare il più grande progetto di collaborazione nella storia del mondo. La filosofia dell'open source che lo sosteneva era semplice: l'Informazione (in questo caso il codice sorgente o le istruzioni di base di un sistema operativo) dovrebbe essere libera, e condivisa gratuitamente per chiunque sia interessalo a migliorarla. Ma anche questi miglioramenti dovrebbero essere condivisi gratuitamente. É un concetto che ha sostenuto secoli di scoperte scientifiche. Ora stava trovando casa nel mondo delle imprese ed era possibile immaginare il suo potenziale come cornice per creare il meglio in ogni campo, dalle strategie legali alle opere liriche.*

*Qualcuno intravide il futuro di questo sistema e ciò che vide non gli piacque. Il faccione occhialuto di Linus divenne uno dei bersagli più diffusi per il gioco delle freccette alla Microsoft, che si trovava finalmente di fronte per la prima volta un vero concorrente. Ma la gente voleva sapere qualcosa di più sul ragazzo da cui era partita tutta questa faccenda,. Il problema era che più Linux e l'*open source *diventavano un successo planetario, meno Linus voleva parlarne. Questo rivoluzionario per caso aveva creato Linux perché giocare con un computer era divertente. (E anche perché le alternative non erano poi tanto attraenti.) Così quando qualcuno provò a convincerlo a parlare a qualche grossa convention dicendogli che milioni dei suoi seguaci volevano almeno* vederlo *in carne e ossa, Linus si offrì di partecipare se avessero permesso al pubblico di tirargli delle torte in faccia a pagamento. Sarebbe stato più divertente, spiegò. Era anche un modo per tirare su dei soldi. Rifiutarono. Non era così, pensavano, che si fa la rivoluzione.*

Rivoluzionari non si nasce. Le rivoluzioni non possono essere programmate. Le rivoluzioni non possono essere gestite.

Le rivoluzioni capitano...

David Diamond

---

[1] Termine gergale americano. Indica sia gli studenti secchioni sia i patiti di informatica.

X-Authentication-Warning: penguin.transmeta.com: torvalds
Owned process doing -bs
Data: Lun, 18 ott. 1999 14:12:27 -0700 (PDT)
Da: Linus Torvalds «torvalds@transmeta.com»
A: David Diamond «ddiamond@well.conm»
Oggetto: Dunque dunque...
MIME-Version : 1.0

Spero che questo sia sempre il tuo indirizzo e-mail. Mi sono accorto di non avere altri modi per contattarti, numeri di telefono o altro, perché probabilmente ho buttato il tuo biglietto da visita insieme a tutti gli altri e perché tu mi hai contattato più per telefono che per e-mail.

Ho pensato molto in questo fine settimana e se ti interessa penso che a me la cosa interessi sempre di più. Facciano un patto: se pensi che possiamo fare un libro divertente, e soprattutto se pensi che ci possiamo divertire a farlo, buttiamoci. Mi puoi portare (insieme alla mia famiglia) In campeggio e (senza la mia famiglia) a fare qualche lancio col paracadute. Tutte cose che sennò non farei perché sono troppo occupato. Dammi una scusa per fare le cose che non ho fatto negli ultimi tre anni anche se ne avevo tutte le possibilità... Probabilmente, una volta finito, non mi metterei a leggere un libro su di me, ma almeno mi ci potrei divertire.

Linus

...e a volte i rivoluzionari ci restano semplicemente incastrati.

Linus Torvalds

Prefazione.
Il senso della vita - PRIMA PARTE
(Il sesso, la guerra, Linux)

Il luogo: questo libro nasce in una Ford Expedition nera nelle corsie che portano verso sud della Statale 5, da qualche parte nella Central Valley della California. Linus e Tove Torvalds, insieme alle loro giovani figlie Patricia e Daniela, sono accompagnati da un infiltrato che farà con loro il viaggio di 560 chilometri fino a Los Angeles, dove visiteranno lo ZOO e un IKEA.

David: C'è una questione fondamentale a cui pensare. È una cosa importante. Di cosa vorresti parlare in questo libro?

Linus: Be', voglio spiegare il senso della vita.

Tove: Linus, ti sei ricordato di fare benzina?

L: Io ho una teoria sul senso della vita. Nel primo capitolo possiamo spiegare ai lettori qual è il senso della vita. Così li agganciamo. E una volta che sono agganciati e si sono comprati il libro, possiamo riempire il resto con stronzate a caso.

D: In effetti come piano non è male. Una volta qualcuno mi ha detto che sin dagli albori dell'uomo ci sono state due domande senza risposta. Uno: «Qual è il senso della vita?» e Due: «Cosa me ne faccio alla fine della giornata di tutte quelle monetine che mi si accumulano in tasca?»

L: Io conosco la risposta alla prima.

D: E quale sarebbe questa risposta?

L: È fondamentalmente breve e dolce. Non darà alcun significato alla tua vita, ma ti dice cosa ti succederà. Allora: ci sono tre cose importanti nella vita. Sono i fattori che motivano qualsiasi cosa tu faccia, per qualsiasi cosa chiunque faccia. Il primo è la sopravvivenza, il secondo l'ordine sociale e il terzo il divertimento. Tutto, nella vita, procede in quest'ordine. E dopo il divertimento non c'è nient'altro. Quindi da un certo punto di vista questo vuol dire che il senso della vita è raggiungere la terza fase. E una volta raggiunta la terza fase, hai finito. Ma prima devi passare attraverso le altre due.

D: Mi sa che ti dovrai spiegare un po'meglio...

Patricia: Papà, possiamo fermarci a prendere un gelato al cioccolato? Ho voglia di gelato al cioccolato...

T: No, tesoro. Devi aspettare. Quando ci fermiamo a fare pipì potrai prendere un gelato.

L: Ti farò qualche esempio. Quello più ovvio è il sesso. É iniziato come sopravvivenza, poi è diventato una questione sociale. É per questo che ci si sposa. E poi diventa divertimento.

P: Allora devo fare pipì.

D: In che senso divertimento?

L: Ok, sto parlando con la persona sbagliata, cambiamo esempio...

D: No, no, torniamo al sesso.

L: E anche su un altro livello...

D: *(tra sé e sé):* Ah, divertimento nel senso che ti diverti quando lo fai, adesso ho capito...

L: ...su un altro livello, dicevo, se pensi al trucco del sesso in un senso biologico. Com'è iniziato il sesso? Era sopravvivenza. All'inizio non era divertimento. Voleva solo dire unirsi. Va be', lasciamo stare questa storia del sesso...

D: No, no, secondo me potremmo farci un capitolo intero.

L: Prendi un altro esempio, la guerra. É ovvio che sia iniziata come sopravvivenza, perché c'è un tizio tra te e la fonte dell'acqua. Poi inizi a combattere per una moglie. K poi la guerra diventa una questione di ordine sociale, già da molto tempo prima del Medioevo.

D: La guerra come mezzo per stabilire un ordine sociale.

L: Giusto. E anche come mezzo per definire te stesso come parte dell'ordine sociale. A nessuno interessa l'ordine sociale di per sé. A tutti invece interessa il proprio posto in quell'ordine. Non importa se sei un pollo in batteria o un essere umano, da questo punto di vista.

D: E adesso la guerra è divertimento?

L: Infatti.

D: Forse per quelli che la guardano in televisione. Per loro potrebbe essere un divertimento.

L: Videogiochi, giochi di guerra, la CNN. La motivazione di una guerra può essere spesso il divertimento. Ma anche la sua percezione è divertimento. É anche la motivazione per il sesso è spesso il divertimento. Certo, è ancora importante il fattore sopravvivenza, soprattutto se sei cattolico. Ma anche se sei cattolico a volte probabilmente pensi anche all'aspetto divertente della faccenda. Quindi non deve essere per forza divertimento puro e semplice. In qualsiasi cosa un pezzo della motivazione può essere la sopravvivenza, un pezzo l'ordine sociale e il resto il divertimento. Prova a pensare alla tecnologia. La tecnologia è nata per la sopravvivenza. E la sopravvivenza non vuol dire solo sopravvivere, vuol dire sopravvivere meglio. Vuol dire costruire un mulino a vento che tira fuori l'acqua dal pozzo...

D: O il fuoco.

L: Certo. È sempre sopravvivenza, ma non è passato all'ordine sociale e al divertimento.

D: E la tecnologia com'è passata all'ordine sociale?

L: Be', in effetti la maggior parte dell'industrializzazione è stata solo una questione di sopravvivenza, nel senso che dicevamo. Per le auto voleva dire costruire auto più veloci e più belle. Ma poi arrivi alla tecnologia nel senso sociale, che ci porta il telefono. E la televisione, in parte. Molta televisione degli albori era fondamentalmente propaganda. E anche la radio. É per questo che molte nazioni iniziarono a investire nelle radio, per l'aspetto sociale della cosa.

D: Stabilire e mantenere l'ordine sociale...

L: Infatti, ma poi viene superata anche quella fase. Oggi la televisione è evidentemente usata soprattutto per divertimento. E pensa a tutti i telefoni cellulari che ci sono oggi. È fondamentalmente un evento sociale. Ma sta diventando divertimento anche quello.

D: Ma allora qual è il futuro della tecnologia? Abbiamo superato la fase della sopravvivenza e stiamo vivendo quella sociale, giusto?

L: Giusto. La tecnologia, prima, non faceva altro che rendere più facile la vita. Era tutta una questione di arrivare prima in un dato posto, avere prezzi più bassi, avere case migliori eccetera. É la differenza con l'informatica qual è? Cosa cambia il fatto che tutti sono connessi? Cos'altro resta da fare? Certo, si possono migliorare le connessioni, ma non è una differenza sostanziale. Dove ci porta la tecnologia? Secondo me il prossimo grande passo è il divertimento.

D: Tutto finisce per trasformarsi in divertimento...

L: Ma questo spiega anche il successo di Linux, almeno fino a un certo punto. Pensa ai tre fattori motivazionali. Il primo è la sopravvivenza, che chi ha un computer dà per scontata. Di base, se hai un computer vuol dire che ti sei già comprato il cibo e roba del genere. Il secondo è l'ordine sociale, e questa è sicuramente la motivazione che tiene i geek inchiodati alle loro scrivanie.

D: Al Comdex hai detto una cosa molto interessante a proposito dello sviluppo di Linux come un gioco di squadra globale...

L: Linux è un ottimo esempio del perché alla gente piacciono i giochi di squadra, e in particolare del perché vogliono far parte di una squadra.

D: Già, stando seduto davanti a un computer tutto il giorno, alla fine probabilmente hai voglia di sentirti parte di qualcosa, qualsiasi cosa sia.

L: È una questione sociale, come qualsiasi altro sport di squadra. Pensa ai giocatori di una squadra di calcio, in particolare quelle per ragazzini. L'aspetto sociale di Linux è molto, molto importante. Ma Linux è anche divertimento, un tipo di divertimento molto difficile da comprare con i soldi. Il denaro è un fattore motivazionale molto potente quando sei al livello della sopravvivenza, perché è facile comprarsi la sopravvivenza. Ma all'improvviso, quando arrivi al livello del divertimento, i

soldi...

D: I soldi sono inutili?

L: No, non sono inutili, perché chiaramente ti puoi comprare i film, un'auto veloce, delle vacanze. E ci sono un sacco di cose che si possono comprare e che contribuiscono a migliorare la tua situazione.

T: Linus, dobbiamo cambiare Daniela. E Patricia deve fare pipì. E io voglio un cappuccino. Secondo te si troverà una caffetteria Starbucks da queste parti? Dove siamo?

D: (alzando lo sguardo): A giudicare dall'odore direi che siamo dalle parti di King City.

L: E adesso tutto questo è su una scala più grande. Non riguarda solo le persone, riguarda la vita. È come la Legge dell'Entropia. In questa Legge dell'Entropia della Vita tutto si muove dalla sopravvivenza al divertimento, ma questo non significa che su scala locale non si possa tornare indietro, ed è evidente che spesso funziona così. A volte le cose si disintegrano, punto e basta.

D: Ma come sistema, tutto si muove nella stessa direzione...

L: Tutto si muove nella stessa direzione, ma non nello stesso momento. Per cui oggi il sesso ha raggiunto la fase di divertimento, la guerra ci è molto vicina, la tecnologia ci si sta avvicinando. Le cose nuove sono quelle di pura sopravvivenza. Come i viaggi nello spazio: a un certo punto - almeno si spera - saranno una questione di sopravvivenza, poi di ordine sociale e poi di divertimento. Pensa al culto della civiltà: segue lo stesso schema. La civiltà inizia come sopravvivenza. Ci si mette insieme per sopravvivere meglio e si costruisce una struttura sociale. Ma alla fine la civiltà esiste per il puro divertimento. Be', forse non proprio puro. E non deve essere necessariamente un divertimento negativo. I Greci avevano un ordine sociale molto forte e allo stesso tempo un sacco di divertimento. Avevano i migliori filosofi della loro epoca.

D: Va bene, ma questo come si collega con il senso della vita?

L: Non molto, in realtà... Ci dice solo che... Be', questo è un po'un problema.

D: A questo collegamento dovrai pensarci un po'.

P: Mamma, guarda le mucche.

L: Dunque: se sai che la vita è basata su questa progressione, ovviamente il tuo scopo nella vita è di completarla. E la progressione non è *una sola* progressione. Tutto ciò che fai fa parte di *molte* progressioni. Si potrebbe tradurre in: «Cosa posso fare per rendere migliore la società?» Tu sai di far parte di una società. Sai che la società si sta muovendo in questa direzione. E allora puoi contribuire a questa progressione.

T: (tappandosi le narici): Che puzza schifosa!

L: Quindi la conclusione è che alla fine siamo tutti qui per divertirci. E allora faremmo meglio a metterci comodi, rilassarci e godercela.

D: Solo per divertirci?

Rivoluzionario per caso

1 La nascita di un nerd

I.

Ero un brutto bambino.

Cosa ci posso fare? Se un giorno a Hollywood faranno un film su Linux, per interpretare il protagonista prendemmo qualcuno tipo Tom Cruise, ma nella versione non hollywoodiana le cose non stanno esattamente così.

Non fraintendetemi. Non è che sembrassi il Gobbo di Notre Dame. Immaginatevi dei grossi dentoni, in modo che chiunque veda una foto della mia infanzia non capisca bene se quello che ha di fronte è un bambino o un castoro. Immaginatevi anche una totale mancanza di gusto nel vestire, abbinata al tradizionale naso extralarge dei Torvalds: cominciate a farvi un quadro della situazione?

A volte mi dicono che un naso grosso dà personalità. E - perlomeno nella mia famiglia - si dice che le dimensioni lei naso siano proporzionate ad altre cose. Ma andate a dirlo a un ragazzino. Per lui il naso serve solo a nascondere i denti. L'immagine dei profili di tre generazioni di Torvalds è un doloroso memento del fatto che - ebbene si - in noi c'è più naso che uomo. O almeno così mi sembrava all'epoca.

E ora passiamo ai dettagli. Capelli castani (quelli che qui negli Stati Uniti chiamano biondi, ma in Scandinavia sono solo castani), occhi azzurri e una leggera miopia che rende sensato l'uso di un paio d'occhiali. E dato che gli occhiali distolgono l'attenzione dal naso, io li porto. Sempre.

Ah, ho già parlato del mio gusto atroce per gli abiti, vero? Il colore di base è il blu, il che di solito vuol dire blue jeans con un dolcevita blu. O magari azzurrino. O quello che capita. Per fortuna la mia famiglia non aveva il pallino della fotografia. Meno prove compromettenti.

Qualche fotografia però è rimasta. In una di queste foto ho più o meno tredici anni e sono in posa con mia sorella Sara, di sedici mesi più giovane di me. Lei è carina. Io sono una visione imbarazzante: un ragazzino pallido e tutto pelle e ossa che si contorce davanti al fotografo, che probabilmente era mia madre. Probabilmente realizzò quel capolavoro mentre usciva di casa per andare a lavorare come traduttrice per la Finnish News Agency.

Essendo nato alla fine dell'anno, il 28 di dicembre, sono sempre il più giovane della classe. Il che vuol dire anche il più piccolo. Più tardi avere sei mesi meno della maggior parte dei compagni di classe non ha grande importanza. Ma alle elementari vuol dire molto.

E sapete una cosa? Tutto questo, il fatto che fossi una specie di castoro con gli

occhiali, con i capelli scarmigliati quasi tutti i giorni (e molto scarmigliati per il resto del tempo) e dei vestiti orribili non significava granché. Perché io avevo una personalità affascinante.

Neanche per idea.

No, guardiamoci in faccia. Ero un nerd. Uno sfigato. Fin da piccolo. Non mi aggiustavo gli occhiali con lo scotch, ma avrei anche potuto farlo perché avevo tutte le altre caratteristiche.

Bravo in matematica, bravo in fisica e privo di qualsiasi capacità sociale. Il tutto molto prima che essere un *nerd* diventasse di moda.

É probabile che abbiate avuto tutti quanti un compagno di scuola come me. Il genietto della matematica, quello che ha i logaritmi nelle vene. Io ero così.

Ma lasciatemi aggiungere qualche altro dettaglio, prima che cominciate a sentirvi troppo dispiaciuti per me. Ero un *nerd,* è vero, e anche uno sfigato, ma non era tutto un disastro. Non ero esattamente atletico ma nemmeno un incapace totale. Il gioco in voga durante la ricreazione era il «brännboll», un gioco di abilità e velocità in cui due squadre cercano di decimarsi tirando una palla. E anche se non sono mai stato il più bravo di tutti, ero comunque tra i primi a essere scelti.

Da un punto di vista sociale ero un *nerd,* è vero, ma a scuola le cose andavano abbastanza bene. Prendevo buoni voti senza dover faticare più di tanto. Non proprio voti altissimi, anche perché non faticavo *affatto.* E avevo un posto accettabile nell'ordine sociale. Nessuno sembrava badare troppo al mio naso, probabilmente perché erano tutti molto più occupati a badare ai propri problemi.

A ripensarci oggi, la maggior parte dei bambini sembrava avere un gusto abbastanza orribile per i vestiti. Crescendo arriva all'improvviso qualcun altro che prende per noi questo tipo di decisioni. Nel mio caso a farlo sono gli uffici marketing delle società informatiche, quelli che scelgono le magliette e i giubbotti da regalare alle conferenze. Oggi mi vesto quasi esclusivamente con vestiti omaggio, cosi non mi tocca di sceglierli. E ho una moglie che decide come completare il mio guardaroba, che sceglie roba tipo i sandali e le calze. E io non me ne devo più preoccupare.

Nel frattempo - dietro al mio naso - sono cresciuto. Adesso posso dire che sono più uomo che naso.

II.

Probabilmente non vi sorprenderà sapere che alcuni dei miei primi e più felici ricordi sono legati ai momenti in cui giocavo con la vecchia calcolatrice elettronica di mio nonno.

Era il padre di mia madre, Leo Waldemar Tornqvist, professore di statistica all'università di Helsinki. Ricordo che mi sono divertito un sacco a calcolare il seno di numeri a caso. Non che mi interessasse particolarmente la risposta (a chi interessa, in fondo?), ma era molto tempo fa e i calcolatori non si limitavano a *darti* la risposta. La *calcolavano.* E mentre lo facevano lampeggiavano un sacco, soprattutto per darti l'idea che «Sì, sono ancora vivo, ci metterò dieci secondi a fare questo calcolo e nel

frattempo lampeggerò per mostrarti quanto sto lavorando.

Era affascinante. Molto più eccitante di un moderno calcolatore che non batte ciglio per calcolare una cosa semplice come il seno di un numero. Con quegli strumenti primitivi sapevi che quello che facevano era *difficile*. Erano loro a dirtelo.

Non ricordo esattamente la prima volta che vidi un computer, ma dovevo avere più o meno undici anni. Era all'incirca il 1981 quando mio nonno comprò un Commodore VIC-20. Dato che passavo così tanto tempo a giocare con la sua calcolatrice magica, dovevo essere supereccitato all'idea di trastullarmi con il nuovo computer. Ma proprio non riesco a ricordarmelo. In effetti non mi ricordo nemmeno quando iniziai ad appassionarmi di computer. Iniziò lentamente e mi crebbe dentro.

Il VIC-20 era uno dei primi computer pensati per un uso domestico. Non richiedeva alcun assemblaggio. Lo attaccavi alla televisione, lo accendevi e te la trovavi lì: una scritta in enormi lettere maiuscole che diceva «READY» e se ne stava in cima allo schermo con un grosso cursore lampeggiante che aspettava un tuo comando.

Il problema era che con quell'aggeggio non c'era granché da fare. Soprattutto all'inizio, quando l'infrastruttura dei programmi commerciali non era ancora nata. La sola cosa che ci si potesse fare era programmarlo in BASIC. Che fu esattamente ciò che il nonno iniziò a fare.

Mio nonno vedeva questo nuovo aggeggio soprattutto come un giocattolo, ma anche come una calcolatrice all'ennesima potenza. Non solo poteva calcolare il seno di un numero molto più in fretta della vecchia calcolatrice elettronica, ma potevi anche dirgli di continuare a farlo automaticamente. E poi il nonno poteva farci a casa molte delle cose che aveva fatto con i grandi computer dell'università.

E voleva che io condividessi con lui quest'esperienza. Stava cercando di farmi appassionare alla matematica.

Così mi sedevo sulle sue ginocchia e digitavo i programmi che lui aveva scritto su carta perché non si trovava a proprio agio con i computer. Non so quanti altri preadolescenti se ne stessero seduti nella stanza del nonno a imparare come semplificare espressioni matematiche e come digitarle correttamente in un computer: io lo facevo. Non ricordo a cosa servissero quei calcoli e non credo di aver saputo minimamente cosa stessi facendo, ma ero lì ad aiutare mio nonno. Probabilmente ci mettevamo più tempo di quanto ne avrebbe impiegato se avesse fatto tutto da solo, ma chi può dirlo? Presi confidenza con la tastiera, una cosa che il nonno non riuscì mai a fare. Lo aiutavo dopo la scuola oppure ogni volta che la mamma mi lasciava a casa dei nonni.

Così iniziai a leggere i manuali del computer e digitare i programmi di prova. Erano esempi di semplici giochi che potevi programmarti da solo. Se facevi tutto per bene alla fine ti trovavi con un tizio che attraversava lo schermo, in una grafica orribile. A quel punto potevi fare qualche cambiamento e far camminare quel tizio attraverso lo schermo con dei colori differenti. Era una cosa che facevi *accadere* tu.

È la sensazione più inebriante di tutte.

Iniziai a scrivere i miei programmi. Il primo programma che scrissi fu lo stesso programma con cui iniziano tutti quanti:

```
10 PRINT "CIAO"
20 GOTO 10
```

Questo programmino fa esattamente quello che vi immaginate: fa apparire la scritta «CIAO» sullo schermo. All’infinito. O almeno finché vi annoia tanto da interromperlo.

Ma è il primo passo. Qualcuno si ferma lì. Per loro è un esercizio stupido: chi mai vorrebbe scrivere «CIAO» un milione di volte sullo schermo? Ma era sempre il primo esempio nei manuali di quei primi giorni dell’informatica casalinga.

E la cosa magica era che potevi cambiarlo. Mia sorella racconta che feci una seconda versione di questo programma che invece di limitarsi a scrivere «CIAO» faceva apparire all’infinito sullo schermo la scritta «SARA È LA MIGLIORE:». In generale non è che fossi il più affettuoso dei fratelli e quel gesto la impressionò molto.

Non ricordo di averlo fatto. Quando scrivevo un programma lo dimenticavo subito e passavo al successivo.

III.

Lasciate che vi parli un po’della Finlandia. Verso ottobre il cielo assume una sgradevole tonalità di grigio e sembra che stia per piovere o per nevicare. Ti svegli tutti i giorni con questa uggiosa sensazione di attesa. La pioggia sarà gelida e spazzerà via ogni ricordo dell’estate. La neve ha la proprietà magica di far luccicare il mondo e di stendere su tutto quanto uno strato di ottimismo. Il problema è che l’ottimismo dura più o meno tre giorni, ma la neve resta per mesi e mesi di un freddo spaccaossa.

A gennaio, se decidi di uscire, te ne vai in giro in una specie di vinavil emotivo. E una stagione di abiti umidi e pesanti e di scivolate sul campo da hockey che hanno creato inondando il cortile della scuola che attraversi per arrivare più in fretta all’autobus. Per le strade di Helsinki l’inverno vuol dire scansare delle matrone imbacuccate che probabilmente a settembre erano simpatiche nonnine ma alle 11 del mattino di un martedì di gennaio ondeggiano sul marciapiede dopo una colazione a base di vodka. E chi può biasimarle? Tra poche ore tornerà il buio e non c’è granché da fare. Ma c’era un sport che mi aiutava a superare l’inverno: la programmazione.

Morfar (il termine svedese per «nonno materno) c’è molto spesso, ma non sempre. Non gli dispiace che tu sieda nella sua stanza quando è fuori. Chiedi ai tuoi i soldi per il tuo primo libro di informatica. È in inglese e devi decodificare il linguaggio. È difficile capire la letteratura tecnica in una lingua che non conosci bene. Usi la tua paghetta per comprarti delle riviste di computer. Una di queste contiene un programma per il codice Morse. La cosa strana di questo programma è che non è scritto in linguaggio BASIC: è una lista di numeri che potrebbero essere tradotti a mano in linguaggio macchina, gli zero e gli uno leggibili dal computer.

Così scopri che in realtà il computer non parla in BASIC. Funziona con un

linguaggio molto più semplice. I ragazzi di Helsinki giocano a hockey e vanno a sciare con i genitori. Tu stai imparando come funziona *effettivamente* un computer. Non sai che esistono dei programmi per tradurre i numeri leggibili dall'uomo negli zero e negli uno comprensibili dal computer, per cui inizi a scrivere i programmi in forma numerica e fai le conversioni a mano. Questo vuol dire programmare in linguaggio macchina, e facendolo inizi a fare delle cose che prima non credevi nemmeno possibili. Riesci a utilizzare meglio il computer. Puoi controllare ogni minimo dettaglio. Inizi a pensare a come potresti fare quelle cose in modo più veloce e in meno spazio. Non ci sono diaframmi di astrazione tra te e il computer, ci sei molto vicino. È l'intimità con una macchina.

Hai dodici anni, tredici, quattordici eccetera. Gli altri ragazzi sono fuori a giocare a calcio. Il computer di tuo nonno è più interessante. La sua macchina è un mondo a sé, in cui è la logica a dettare legge. In classe ci saranno più o meno tre persone con un computer e solo una lo usa per le tue stesse ragioni. Vi incontrate una volta la settimana. É l'unica attività sociale sulla tua agenda, a parte qualche nottata passata al computer.

E non ti dispiace. É divertente.

Questo dopo il divorzio. Papà vive in un'altra zona di Helsinki. Pensa che suo figlio non dovrebbe avere un unico interesse, così ti iscrive a pallacanestro, il suo sport preferito. É un disastro. Sei lo sfigato della squadra. Dopo una stagione e mezza usi le parolacce più ignobili per dirgli che lasci la squadra, che la pallacanestro è il suo sport, non il tuo. Il tuo nuovo fratellastro, Leo, è più atletico. Poi diventerà luterano, come l'86 per cento della popolazione finlandese. E a quel punto che papà, l'agnostico di ferro, capisce che forse è stato un pessimo genitore (un sospetto che l'aveva sfiorato già anni prima, quando Sara si era convertita al cattolicesimo).

Il nonno con il computer non è esattamente uno spasso. Perde i capelli, è un po'sovrappeso. È il tipo di professore con la testa fra le nuvole, difficile da avvicinale. Non è una persona estroversa. Immaginate un matematico che se ne sta a pensare con lo sguardo fisso nel vuoto, senza dire una parola. Non potete nemmeno immaginare cosa sta pensando. Qualche astrusa formula di calcolo? Alla vicina, la signora Sammalkorpi? Sono anch'io così: mi impallo. Quando sto seduto davanti al computer divento irritabile se qualcuno mi disturba. Provate a chiederlo a Tove.

I miei ricordi più vividi di Morfar non hanno a che fare con il suo computer ma con il suo piccolo cottage rosso. A Helsinki molta gente aveva un posticino per l'estate, magari anche solo un'unica stanza di otto metri per otto. Queste casette stanno su piccoli appezzamenti di terra, più o meno quaranta metri per quaranta, e la gente ci va a fare giardinaggio. Di solito hanno un appartamento in città e poi questo posto dove coltivano le patate o le mele o dei roseti. Di solito si tratta di gente anziana, perché i giovani sono occupati a lavorare. Queste persone diventano tremendamente competitive a proposito di ciò che coltivano. Fu proprio al cottage che lui piantò il mio melo, un alberello molto giovane. Forse c'è ancora, a meno che non sia cresciuto tanto da spingere qualche vicino invidioso ad abbatterlo di nascosto.

Quattro anni dopo avermi fatto conoscere i computer, Morfar ha un ictus cerebrale e resta semiparalizzato. É uno shock per tutti. Sta in ospedale per circa un anno ed è tutta la famiglia che hai, ma la cosa non ti tocca più di tanto. Forse è una difesa o

forse è solo perché da giovani si è insensibili. Lui non è più lo stesso e a te non piace andarlo a trovare. Ci vai al massimo un paio di volte al mese. Tua madre ci va più spesso. Tua sorella, che ha assunto da molto tempo il ruolo di crocerossina della famiglia, è la più assidua.

Quando muore, la macchina viene a vivere con te. Nessuna discussione al riguardo.

IV.

Facciamo un passo indietro.

Magari oggi la Finlandia è la nazione più fighetta del mondo, ma qualche secolo fa era solo uno scalo tecnico per i vichinghi diretti a Costantinopoli. Poi i vicini svedesi decisero di pacificare i finlandesi e mandarono in missione il vescovo Enrico, d'origine inglese, che arrivò come missionario cattolico nell'anno 1155. Gli svedesi utilizzarono le fortezze finlandesi per tenere lontani i russi e finirono per vincere la corsa al controllo contro l'impero dell'Est. Per popolare di svedesi la colonia, a chi vi si trasferiva vennero offerte terre e agevolazioni fiscali. Gli svedesi fecero il bello e il cattivo tempo fino al 1714, quando la Russia prese il comando per un intervallo di sette anni. Poi gli svedesi si ripresero la colonia fino al 1809, quando la Russia e Napoleone attaccarono la Finlandia, che restò sotto il controllo russo fino alla Rivoluzione Comunista del 1917. Nel frattempo i discendenti dei primi coloni svedesi hanno generato gli odierni 350.000 finlandesi di lingua svedese, un gruppo che rappresenta attualmente il 5 per cento circa della popolazione.

Compresa la mia bislacca famiglia.

Il mio bisnonno materno era un contadino relativamente povero di Jappo, un paesino vicino alla città di Vasa. Aveva sei figli, due dei quali si laurearono. Il che la dice lunga sulle prospettive di avanzamento sociale in Finlandia. Sì, ci si stufa a morte del buio invernale, è una noia togliersi le scarpe ogni volta che si entra in una casa, ma si può avere un'istruzione universitaria gratis. Una bella differenza da quanto accade negli Stati Uniti, dove moltissimi ragazzi crescono con un senso di disperazione. Uno di quei sei figli era mio nonno, Leo Waldemar Tornqvist, l'uomo che mi fece conoscere i computer.

Poi c'era il mio nonno paterno. Fu lui a inventarsi il cognome Torvalds. Lo creò a partire dal suo secondo nome di battesimo. Si chiamava Ole Torvald Elis Saxberg. Era nato senza un padre (Saxberg era il cognome da signorina di sua madre) e prese il cognome di Karanko dal gentiluomo che in seguito la mia bisnonna sposò. A Farfar («nonno paterno») non piaceva quel tizio, tanto che alla fine, nel 1937, si cambiò cognome. Fece cancellare quello vecchio e aggiunse una s a Torvald perché avesse più il suono di un cognome. Torvald (senza la s) significa «il dominio di Thor». Avrebbe fatto meglio a trovarsi un altro cognome, perché quella s finale distrugge il significato del nome d'origine, confondendo sia chi parla svedese sia chi parla finlandese e lasciando tutti nel dubbio su come pronunciarlo. E di solito pensano che si scriva Thorwalds. Ci sono ventuno Torvalds nel mondo: siamo tutti parenti e abbiamo tutti gli stessi problemi con il nostro cognome.

Forse è per questo che sulla Rete io sono sempre semplicemente «Linus». Con «Torvalds» si fa troppa confusione.

Questo nonno non insegnava all'università. Faceva il giornalista e il poeta. Il suo primo lavoro fu quello di caporedattore nel quotidiano di un paesino a un centinaio di chilometri da Helsinki. Lo licenziarono perché sul lavoro alzava il gomito con una certa regolarità. Il matrimonio con mia nonna andò a rotoli. Si trasferì a Turku, nel sud-ovest della Finlandia, dove si risposò, divenne caporedattore del quotidiano locale e pubblicò diversi libri di poesia. I problemi con l'alcol non li superò mai. Noi andavamo a trovarlo per Natale e per Pasqua, e facevamo visita anche a mia nonna. Farmar Marta vive a Helsinki, dove è nota per le sue incredibili torte.

Farfar è morto cinque anni fa.

Lo ammetto: non ho mai letto un suo libro. É un fatto che mio padre non manca di far notare a qualsiasi sconosciuto. Il giornalismo è un po'una tradizione di famiglia. La leggenda vuole che uno dei miei bisnonni, Ernst von Wendt, giornalista e romanziere che parteggiava per i Bianchi, sia stato arrestato dai Rossi durante la guerra civile finlandese che seguì l'indipendenza dalla Russia nel 1917. (Okay, mi avete beccato, non ho mai letto nemmeno i suoi libri, ma mi dicono che non mi sono perso un granché.) Mio padre Nils (che tutti conoscono come Nicke) è un giornalista televisivo e radiofonico ed è stato un attivista del Partito Comunista sin dagli anni Sessanta, quando era studente universitario. Le sue idee politiche si fecero più radicali quando studiò alcune delle atrocità commesse in Finlandia contro chi simpatizzava per il comunismo. Oggi, a decenni di distanza, ammette l'ingenuità all'origine del suo entusiasmo per il comunismo. Incontrò mia madre Anna (detta Mikke) negli anni Sessanta, quando entrambi erano studenti contestatori. Erano in gita con un club di studenti di lingua svedese di cui lui era presidente. C'era un altro ragazzo che stava dietro a mia madre, e mio padre – mentre si preparavano a tornare in autobus a Helsinki - gli disse badare che tutti i bagagli venissero caricati. Dopodiché colse l'occasione per accaparrarsi il posto accanto alla madre e convincerla a uscire con lui. E poi dicono che io sono il genio di famiglia!

lo sono nato tra una marcia di protesta e un'occupazione universitaria, probabilmente con in sottofondo un disco Joni Mitchell. Il nido d'amore della nostra famiglia era una stanza nella casa dei miei nonni. La mia prima culla fu Il cestino per la biancheria. Grazie a Dio non ricordo nulla di quel periodo. A un certo punto, quando avevo più o Inno tre anni, mio padre accettò di fare il servizio militare di undici mesi anziché andare in prigione come - da buon obiettore di coscienza - avrebbe certo preferito. Divenne un soldato tanto bravo e un tiratore tanto preciso che spesso gli venivano accordati dei fine settimana di licenza. La mitologia di famiglia vuole che mia sorella Sara sia stata concepita durante una di quelle visite. Mia madre, quando non scorrazzava in giro i suoi due mostriciattoli biondi, lavorava come redattrice all'ufficio esteri della Finnish News Agency. Oggi fa la *graphic editor*.

È un miracolo che io sia scampato a questa minidinastia giornalistica. Sara fa la *freelance,* traduce articoli per i notiziari e lavora a sua volta per la Finnish News Agency. Il mio fratellastro Leo è un maniaco del video e vorrebbe fare il regista cinematografico. Dato che i membri della mia famiglia sono praticamente tutti giornalisti, mi sento autorizzato a scherzare con i reporter su quanto siano delle

schifezze di esseri umani. Lo so che quando dico cose del genere faccio la figura dello stronzo, ma nel corso degli anni la nostra casa in Finlandia ha ospitato una buona quantità di giornalisti che non si fermavano davanti a nulla per avere delle storie da raccontare, o che se le inventavano di sana pianta, sempre con l'aria di avere appena alzato un po'troppo il gomito. Ok, mi correggo, di averlo alzato *decisamente* troppo.

Erano i momenti in cui mi rifugiavo in camera mia. Come quando mamma aveva la luna storta. La scena è questa: abitiamo in un bilocale al secondo piano di un palazzo giallino senza infamia e senza lode sulla Stora Robertsgatan, a Rödbergen, un piccolo quartiere vicino al centro di Helsinki. Sara e il suo insopportabile fratello (maggiore di sedici mesi) dividono la stessa camera da letto. C'è un parchetto lì vicino, ha preso il nome dalla famiglia Sinebrychoff, proprietaria di una distilleria locale. È una cosa che mi è sempre sembrata strana, ma che differenza c'è rispetto a dare a uno stadio di baseball il nome di un rivenditore di prodotti per ufficio? (Dato che una volta ci avevamo visto un gatto, il Sinebrychoffsparken dalla mia famiglia veniva chiamato «Catpark».) Lì c'è una casetta bianca vuota, dove si riuniscono i piccioni. Il parco è costruito su una collinetta e in inverno ci si può andare con lo slittino. Un altro posto per giocare è il cortile di cemento dietro il nostro palazzo, oppure il condominio stesso. Quando giochiamo a nascondino ci divertiamo ad arrampicarci sulla scala antincendio per cinque piani, fino al soffitto.

Ma niente era divertente come il computer. Con il computer a casa potevo stare alzato tutta la notte. Tutti i ragazzi stanno alzati a leggere «Playboy» sotto le lenzuola. Solo che io, invece di leggere «Playboy», facevo finta di dormire, aspettavo che mamma si allontanasse, mi alzavo e mi sedevo davanti al computer. Non era ancora arrivata l'epoca delle *chatroom.*

«Linus, è pronto da mangiare!» La maggior parte delle volte non esci nemmeno dalla tua stanza. Poi tua madre inizia dire alle sue amiche giornaliste che sei un bambino proprio facile da tirare su, basta chiuderti in un bugigattolo buio con un computer e ogni tanto tirarti dentro un po'di pasta cruda. Non è molto lontana dalla verità. Nessun pericolo di venir rapito, per questo ragazzino. (E poi... come dire... se ne sarebbero accorti?) In effetti i computer erano meglio per i ragazzini quando erano meno sofisticati, quando anche degli sbarbatelli come me potevano mettersi a trafficare sotto il cofano. Oggi i computer hanno lo stesso problema delle auto: a mano a mano che diventano più i complessi, per le persone è sempre più difficile smontarli e montarli e quindi imparare qualcosa su di loro. Quando è stata l'ultima volta che avete fatto alla vostra macchina qualcosa di più complesso che cambiare il filtro dell'olio?

Invece di trafficare sotto i cofani metaforici dei loro computer, i ragazzini di oggi si spappolano il cervello con troppi videogiochi. Non che ci sia qualcosa di sbagliato nei giochi. Sono stati tra i primi programmi che ho creato.

Ce n'era uno in cui controllavi un piccolo sottomarino in una grotta acquatica. È un concetto di base dei videogiochi. Il mondo si sposta orizzontalmente, scorre, e tu sei in un sottomarino e devi evitare di urtare le pareti e i mostri marini. La sola cosa che in realtà si muove è il mondo. I pesci si muovono con il mondo. A mano a mano che vai avanti con il gioco, tutto inizia a muoversi sempre più veloce. Nel frattempo la

grotta si fa sempre più stretta. Non puoi vincere in un gioco del genere, ma non era questo il punto. É divertente giocare per una settimana o giù di lì e poi passare al gioco successivo. Il punto è soltanto scrivere il codice che fa accadere tutto questo.

Ci sono altri giocattoli, modellini d'aerei e di navi e di auto e di ferrovie. A un certo punto papà compra un trenino tedesco che costa un sacco di soldi. Il ragionamento era questo: da piccolo non aveva mai avuto un trenino e avrebbe potuto diventare uno di quegli hobby papà-figlio. É divertente, ma niente che possa anche solo avvicinarsi alla sfida di un computer. L'unica volta in cui ti viene tolto il computer non è perché ci passavi sopra troppo tempo, ma come punizione per qualcos'altro, tipo avere fatto a botte con Sara. Per tutte le medie e le superiori voi due siete sempre in competizione, soprattutto per quanto riguarda i voti scolastici.

La competizione porta sempre buoni risultati. Senza le mie continue critiche, Sara non sarebbe mai stata abbastanza motivata da scrivere sei tesine di fine anno invece delle cinque necessarie per diplomarsi in Finlandia. D'altra parte Sara va ringraziata per il fatto che il mio inglese non sia del tutto atroce. Mi prendeva sempre in giro per il mio inglese, che per anni è stato il tipico anglofinlandese. È stato questo a farmi migliorare. E se è per questo anche mia mamma mi prendeva in giro, ma soprattutto perché mostravo poco interesse per le mie compagne di classe che volevano prendere lezioni dal «genio della matematica:».

In certi periodi vivevamo con mio papà e la sua fidanzata, in altri Sara viveva con mio papà e io con mia mamma. In altri ancora vivevamo entrambi con la mamma. Tra l'altro, in svedese non esiste un termine per indicare una «famiglia disfunzionalo. Non avevamo molti soldi a causa del divorzio. Uno dei miei ricordi più vivi è quello di quando mia mamma doveva portare al banco dei pegni il suo unico investimento, una quota di azioni dell'azienda dei telefoni di Helsinki che ti veniva data quando stipulavi un contratto telefonico. Probabilmente valeva qualcosa come 500 dollari e ogni tanto, quando le cose andavano particolarmente male, lei doveva portare la matrice al banco dei pegni. Ricordo di essere andato con lei una volta e di essermi sentito in imbarazzo. (Oggi faccio parte del consiglio d'amministrazione di quella società. In effetti la Helsinki Telephone Company è l'unica società di cui sono consigliere.) Mi sentii in imbarazzo anche quando, dopo che avevo messo via quasi tutti i soldi necessari per comprarmi il mio primo orologio, la mamma volle che chiedessi al nonno di metterci il resto.

Ci fu un periodo in cui la mamma lavorava di notte, così io e Sara dovevamo arrangiarci per la cena. Teoricamente avremmo dovuto andare al negozio all'angolo e comprare del cibo da far mettere in conto. Noi invece ci compravamo delle caramelle ed era bellissimo stare alzati fino a tardi con il computer. In circostanze simili gli altri ragazzi avrebbero «letto «Playboy» sotto le lenzuola.

Poco dopo l'infarto di mio nonno, Mormor non fu più autosufficiente. Venne ricoverata in una casa di riposo dove rimase per dieci anni. Come diceva lei, era «rintontita:». Dopo due anni che si trovava nella casa di riposo, noi ci trasferimmo nel suo appartamento. Era al primo piano di un vecchio edificio solido di epoca russa sulla Petersgatan, vicino al grazioso parco che costeggia il porto di Helsinki. C'erano un cucinino e tre camere da letto. Sara si prese quella grande. Io, da buon adolescente a cui bastavano uno sgabuzzino buio e una pastasciutta ogni tanto, mi trasferii In

quella più piccola. Appesi dei teli scuri alle finestre in modo che la luce del sole non potesse entrare. Il computer trovò posto su una piccola scrivania messa contro la finestra, a non più di mezzo metro dal mio letto.

Sapevo ben poco di Linus Torvalds quando un redattore del «San Jose Mercury News» mi chiese di scrivere un suo profilo nella primavera del 1999. Linux era diventato una specie di parola d'ordine dalla primavera precedente, quando una serie di aziende a partire da Netscape ne avevano adottato l'idea di codice open source o addirittura il sistema operativo. Ma non è che fossi stato su Marte fino ad allora. All'inizio degli anni Novanta avevo lavorato come redattore in una rivista che si occupava di argomenti come Unix e l'open source, per cui da qualche parte nel mio cervello c'era una qualche scheda impolverata a proposito di Linus. E quella scheda mi diceva che era un universitario finlandese che aveva scritto un'eccezionale versione di Unix nella sua stanzetta e l'aveva distribuita gratuitamente su internet. Non era una scheda particolarmente accurata. Il redattore mi aveva telefonato perché Linus era appena stato l'attrazione principale a una recente presentazione di Linux a San Jose, il che aveva spinto la rivista ad assegnarmi quel pezzo con le parole: «Abbiamo una superstar globale proprio qui a... ehm... Santa Clara.» Mi mandarono via fax qualche articolo di giornale.

Linus si era trasferito nella Silicon Valley due anni prima e lavorava per l'allora supersegreta Transmeta Corporation, dove da anni si stava sviluppando un microprocessore che avrebbe dovuto sconvolgere il settore informatico. In qualche modo aveva ottenuto un lavoro che gli consentiva di mantenere la sua posizione assai impegnativa di leader di Linux e autorità finale su qualsiasi cambiamento apportato al sistema operativo. (I suoi seguaci avevano di fatto avviato la manovra legale per dargli la proprietà legale sul marchio Linux.) E aveva anche il tempo per girare il mondo come araldo del nascente movimento per l'open source.

Ma era diventato una specie di misterioso eroe popolare. Mentre Bill Gates, il nemico preferito di tutti, viveva come un re nella sua Xanadu, Linus stava con la moglie e le figlie piccole in un bilocale incasinato di Santa Clara. Sembrava non badare alle enormi ricchezze che stavano piovendo su stormi di programmatori di minor talento. E la sua stessa presenza sollevava un dubbio impronunciabile tra i servi della stock option della Silicon Valley: come era possibile che una persona tanto brillante fosse così poco interessata a fare soldi?

Linus non ha portaborse, non ascolta i messaggi in segreteria telefonica e risponde di rado alle e-mail. Mi ci sono volute settimane per parlargli al telefono, ma una volta raggiunto questo obiettivo ha accettato tranquillamente di fare un'intervista appena gli fosse possibile, e cioè un mesetto dopo: maggio 1999. Avendo sviluppato una passione professionale per mettere i soggetti delle mie interviste in posizioni compromettenti, decisi che una sauna finlandese sarebbe stata uno sfondo perfetto per il mio articolo. In una Mustang decappottabile in affitto e con un fotografo al seguito, ci avviammo verso Santa Cruz e verso quella che ci era stata raccomandata come la migliore sauna della Bay Area, sul terreno di un villaggio vacanze New Age per nudisti.

Lui era armato di una lattina aperta di Coca quando uscì dalla porta degli uffici

della Transmeta, in un anonimo centro direzionale di Santa Clara. Portava la tipica uniforme da programmatore: jeans, maglietta promozionale di una conferenza e l'inevitabile accoppiata calze&sandali che lui sosteneva di avere adottato già da prima di incontrare un altro programmatore. «Deve essere una specie di legge di natura dei programmatori», mi rispose quando gli chiesi qualcosa in proposito.

La prima domanda che feci a Linus sul sedile posteriore di quell'auto fu una battuta buttata là. «La tua famiglia si occupa di tecnologia?», gli chiesi giocherellando con il registratore.

«No, fondamentalmente sono tutti giornalisti», rispose lui. E poi aggiunse: «Quindi so che razza di feccia siete.»

Non gliel'avrei lasciata passate tanto facilmente.

«Ah, quindi tu vieni dalla feccia?», risposi. Il miglior programmatore del mondo scoppiò a ridere tanto forte che si mise a tossire e spruzzò della Coca sulla nuca del fotografo-autista. Divenne rosso. Iniziò così un pomeriggio memorabile.

Il seguito fu ancora più bizzarro. I finlandesi vanno matti per le saune e questa era la prima volta che lui ci andava da quasi tre anni. La superstar pallida e nuda con gli occhiali appannati se ne stava seduta sulla panca più alta, con i capelli bagnati che gli si appiccicavano alla faccia e un fiume di sudore che scorreva giù da quella che più tardi - in uno slancio di pura bontà - avrei descritto come «un'incipiente pancetta». Era circondato da tipici abitanti di Santa Clara abbronzati ed egocentrici e dalle loro monotone chiacchiere New Age e sembrava volare sopra tutto questo mentre sottolineava entusiasticamente le caratteristiche autentiche della sauna. Aveva un sorriso beato stampato in faccia.

Sono convinto che gli abitanti della Silicon Valley siano in linea generale più felici di chiunque altro. Intanto si trovano al pannello di controllo della rivoluzione economica. Ma soprattutto stanno diventando tutti intollerabilmente ricchi, sia nella New Valley che nella Old Valley. Ma qui non si vede mai la gente sorridere, per lo meno non fuori dagli uffici dei loro broker.

La maggior parte dei tecnologi famosi - e anche di quelli che non lo sono - hanno un irrefrenabile desiderio di farti sapere quanto sono brillanti. E anche che sono i protagonisti di una missione molto più importante, per dire, della pace nel mondo. Con Linus non era così. In effetti la sua mancanza di Ego sembrava assolutamente disarmante e lo rendeva adorabile al confronto della pomposa élite della Silicon Valley. Linus appariva superiore a tutto questo. Ai fanatici della New Age. Ai miliardari hi-tech. Non sembrava tanto una renna ipnotizzata dai fari globali quanto un adorabile alieno teletrasportato sulla terra per mostrarci la follia delle nostre vite egocentriche.

E avevo la sensazione che non uscisse molto.

Linus aveva detto poco prima che una parte importante del rituale della sauna consisteva nello starsene seduti - dopo essere usciti dai vapori - a bere birra e parlare di questioni mondiali. Per prepararci avevamo nascosto delle lattine di Fosters tra i cespugli. Recuperammo le nostre birre e ci piazzammo nella vasca calda e tranquilla, dove aprimmo le nostre lattine di Fosters e il fotografo ci fece qualche scatto. Scoprii che Linus era inaspettatamente preparato sulla storia dell'economia americana e sulle politiche mondiali. Secondo lui gli Stati Uniti

sarebbero stati avvantaggiati se sia le corporazioni sia i partiti politici avessero adottato l'approccio conciliatorio dei politici europei. Immerse gli occhiali nella vasca calda per pulirli, dicendo che non aveva veramente bisogno di portarli ma che aveva iniziato a metterseli da ragazzino perché gli facevano sembrare più piccolo il naso. Fu a quel punto che apparve un'impiegata vestita di tutto punto e ci ordinò senza il minimo umorismo di consegnarle le nostre birre, che in quell'ambiente rigorosamente analcolico erano considerate come merce di contrabbando.

La nostra unica possibilità era fare la doccia, vestirci e trovare un bar per finire la nostra conversazione. La maggior parte delle persone che incontri nella Silicon Valley hanno una specie di culto per sé stessi. Sono tanto concentrati sul proprio business a sulla propria applicazione rivoluzionaria o sull'Informatica (con la I maiuscola, naturalmente) che non sembra esistere nient'altro. Nulla interrompe il loop incessante di autocelebrazione che ti spacciano come conversazione. E noi invece eravamo lì, seduti al sole in una birreria ad assaggiare vino d'orzo Godawful, con Linus che continuava a chiacchierare come un canarino liberato dalla gabbia, confessava la sua dipendenza dal Rock Classico e da Dean Koontz, rivelava il suo debole per le sitcom più idiote, raccontava segreti di famiglia.

E non aveva troppa voglia di starsene nel giro dei ricchi & potenti. Gli chiesi cosa avrebbe voluto dire a Bill Gates, ma lui non era minimamente interessato a incontrarlo. «Non ci sarebbero grandi punti di contatto», mi spiegò. «Io non sono per niente interessato alla cosa che lui sa fare meglio di chiunque altro al mondo. E a lui non interessa quella che forse io so fare meglio di chiunque altro. Io non gli saprei dare nessun consiglio commerciale e lui non me ne saprebbe dare sulla tecnologia.»

Mentre risalivamo verso Santa Clara una Jeep Cherochee nera ci si accostò e il tizio a bordo urlò «Ehi, Linus!» e tirò fuori una macchina fotografica usa e getta per ritrarre quello che sembrava considerare il suo eroe, seduto sul sedile posteriore della Mustang decappottabile a sorridere al vento.

Mi presentai a casa sua una settimana più tardi, all'ora del bagnetto. Lui ripescò la biondissima figlia di un anno dalla vasca e si mise a cercare un posto in cui depositarla mentre ripescava anche la biondissima figlia di due anni. Mi mise in braccio la piccola e lei si mise immediatamente a strillare. Sua moglie Tove, che per tutto il tempo era stata in un'altra stanza, arrivò in soccorso. È una donna minuta, piacevole e con un cardo tatuato sulla caviglia. Poco dopo ci trovammo tutti a leggere libri di favole svedesi e inglesi per le bimbe. Poi ci trasferimmo in garage, tra scatoloni ancora da disimballare, dove i Torvalds discussero dell'impossibilità di permettersi «una vera casa con un vero giardino» nella Silicon Valley. E lo dicevano senza alcuna amarezza.

Ma la cosa più stupefacente era che non sembravano vedere l'ironia di quella faccenda.

Poi ci mettemmo a guardare Jay Leno in tv, con qualche lattina di Guinness a farci compagnia. Fu a quel punto che capii che avremmo dovuto fare un libro.

V.

Di base restai seduto davanti a un computer per quattro anni.

Ok, c'era la scuola: la Norssen High, il più centrale dei cinque licei in lingua svedese di Helsinki e una delle più vicine a casa mia. La matematica e la fisica erano interessanti e non mi creavano problemi. Ma quando per una materia la era necessario imparare qualcosa a memoria, il mio entusiasmo diminuiva all'istante. Così la storia mi sembrava noiosa quando dovevo imparare la data della Battaglia di Hastings, ma si faceva interessante quando si discuteva dei fattori economici che caratterizzavano un paese. La stessa cosa accadeva per la geografia. Voglio dire, a chi importa davvero sapere quanti sono gli abitanti del Bangladesh? Be', a pensarci bene potrebbe essere una cosa importante per un sacco di gente. Ma il fatto è che per me era molto più facile sognare a occhi aperti il mio computer quando studiavamo qualcosa di più interessante delle statistiche, cose come i monsoni, per esempio, o i motivi per cui si scatenavano.

L'educazione fisica era tutt'altra faccenda. Probabilmente non vi sorprenderà sapere che non ero la persona più atletica della Scandinavia. E all'epoca - che ci crediate o no - ero anche pelle e ossa. Non mi dispiaceva la ginnastica, ma quando c'erano le partite di calcio o di hockey su giaccio per me arrivava il momento di marinare.

E questo si rifletteva sui miei voti. In Finlandia i voti scolastici vanno da 4 a 10. Io prendevo 10 e qualche 9 in matematica, fisica, biologia e tutto il resto, e 7 in educazione fisica. Una volta anche un 6. E presi un 6 anche in falegnameria. Non era il mio forte. Altri ragazzi si portavano a casa dei bei portatovaglioli o degli appendiabiti come souvenir delle lezioni di falegnameria. Tutto quello che io ho conservato di quelle lezioni sono delle schegge di legno che dopo tutti questi anni sono ancora conficcate nel mio pollice. A questo punto dovrei dirvi che è stato mio suocero a costruire le magnifiche altalene che si possono ammirare nel mio giardino e su cui le mie bambine passano un sacco di ore.

Il mio liceo non era una di quelle istituzioni per ragazzi eccezionalmente intelligenti o ambiziosi tanto comuni nelle città americane. Scuole di questo genere non fanno parte del modo di pensare finlandese. Le scuole in Finlandia non separano i bravi studenti dagli asini. Ogni scuola però aveva una sua specialità, una materia non obbligatoria che non avresti potuto studiare in nessun'altra scuola. Nel caso della Norssen High School era il latino. E il latino era divertente. Molto più divertente del finlandese e dell'inglese.

Peccato che sia una lingua morta. Mi piacerebbe trovarmi con qualche amico a raccontare barzellette in latino o magari a discutere le strategie di progettazione dei sistemi operativi.

Mi piaceva anche passare il tempo in un bar vicino alla scuola. Era il ritrovo di quelli che non si nascondevano dietro la scuola a fumare sigarette. Ci potevi andare quando saltavi le lezioni di educazione fisica o quando avevi una pausa di un'ora tra

le lezioni, cosa che accadeva ogni tanto.

Quel posto era un rifugio per secchioni sin dai tempi in cui si usavano i regoli. Ed era anche l'unico bar in cui gli studenti potevano mettere tutto in conto. Ordinavi quello che volevi, loro annotavano quello che avevi mangiato e bevuto e poi, quando in qualche modo mettevi insieme un po'di soldi, pagavi il conto. Conoscendo la mania finlandese per la tecnologia, oggi probabilmente registrano tutto in un database.

La mia ordinazione era sempre la stessa: una Coca e una ciambellina.

Così giovane e già così salutista...

In linea generale io a scuola ero più bravo di mia sorella Sara, che era più socievole, più carina, più affabile e - a questo punto dovrei confessarlo - ha ricevuto l'incarico di tradurre questo libro in svedese. Ma alla fine mi batteva perché studiava più materie. I miei interessi erano più ristretti. Ero conosciuto come il Matematico.

In effetti le uniche volte in cui portavo delle ragazze a casa era quando mi chiedevano di dare loro delle ripetizioni. Non è che sia successo moltissime volte e mai su mia iniziativa, ma mio padre sostiene con una certa fantasia che fossero interessate a qualcosa in più delle lezioni di matematica. (Secondo lui si trattava di adepte della sua famosa equazione Naso di Carattere = Uomo di Carattere.) Ma se anche avessero voluto fare qualcosa di più interessante dei compiti di matematica, non avrebbero trovato in me un partner particolarmente collaborativo. Voglio dire, nemmeno riuscivo a immaginare esattamente cosa volesse dire limonare. Mi era capitato di spremere qualche decina di limoni per preparare qualcosa di fresco da bere d'estate e non mi era poi sembrato un granché, come attività.

Sì, ero decisamente un secchione. Nessun dubbio in proposito. E questo prima che i secchioni iniziassero a essere considerati sexy. Be', forse non proprio sexy. Diciamo alla moda. Io ero sia secchione sia timido, anche se probabilmente *tutti* i secchioni sono timidi.

E così me ne stavo seduto davanti al computer ed ero felicissimo.

Quando ci si diploma al liceo in Finlandia si indossa un buffo cappello bianco con una striscia nera. C'è una cerimonia in cui vengono consegnati i diplomi e quando torni a casa ci trovi tutti i parenti con un sacco di champagne, fiori e una torta. E c'è anche una festa con tutti i compagni di corso in un ristorante. Io mi feci tutta la trafila e immagino di essermi pure divertito, ma non ho particolari ricordi in proposito. Però chiedetemi le specifiche tecniche del mio primo 68008 e ve le potrei snocciolare senza il minimo tentennamento.

VI.

Il mio primo anno di università fu abbastanza produttivo. Riuscii a prendere tutti i crediti necessari (in Finlandia vengono chiamati «settimane di studio). Fu l'unico anno in cui lo feci. Forse fu l'eccitazione del nuovo ambiente, o la possibilità di approfondire gli argomenti, o forse mi sentivo più a mio agio a studiare che a fare vita di società e vomitare sui miei amici con religiosa regolarità. Non so a cosa dare

la colpa per le mie prestazioni dignitose in quel primo anno di università. Ma vi posso assicurare che non accadde mai più. La mia carriera accademica prese decisamente una strada in discesa.

A quel punto non avevo ancora deciso una specializzazione. Alla fine scelsi informatica, con fisica e matematica come indirizzi. Però c'era il problema che in tutta l'Università di Helsinki c'era soltanto un altro studente di lingua svedese che voleva specializzarsi in informatica, Lars Wirzenius. Ci iscrivemmo insieme alla Spektrum, l'associazione degli studenti di scienze di lingua svedese, e finì per rivelarsi una cosa assai divertente. Il club era composto da studenti di scienze dure, tipo fisica e chimica. In altre parole: niente donne.

Ma condividevamo la sala del club con l'organizzazione degli studenti di scienze morbide (biologia, psicologia eccetera) di lingua svedese. Così potevamo interagire con delle ragazze, per quanto per alcuni di noi non fosse un compito facile. Ok, lo ammetto, per *tutti* noi.

La Spektrum aveva molti cerimoniali in comune con le confraternite all'americana, ma non dovevi vivere con gli altri iscritti e non dovevi necessariamente avere a che fare con gente a cui non interessassero le scienze. Tenevamo degli incontri tutti i mercoledì sera, e fu lì che imparai la differenza tra una pilsner e una ale. Una volta ogni tanto si svolgevano delle gare a chi beveva più vodka. Ma perlopiù questo genere di cose iniziò ad accadere dopo che io ebbi terminato la mia carriera universitaria. E sì che io all'università ci sono rimasto un bel po': otto anni, che mi hanno fruttato soltanto un master (senza contare la laurea honoris causa che mi è stata consegnata nel giugno 2000).

Ma il primo anno fu tutta una corsa in tram tra le aule in cui si tenevano le lezioni e la mia stanza da letto, in cui si accumulavano pile di libri e di strumentazione informatica. Me ne restavo a letto a leggere un thriller fantascientifico di Douglas Adams, poi lo buttavo sul pavimento e prendevo un testo di fisica, poi rotolavo fuori dal letto e mi sedevo al computer a scrivere un programma per un nuovo gioco. La cucina era attaccata alla camera da letto e io mi ci trascinavo ogni tanto per farmi un caffè o mangiucchiare qualche biscotto.

Magari tua sorella è lì in giro, o magari è fuori con gli amici. O magari in questo periodo vive con tuo padre. Magari tua madre è lì o magari sta lavorando o magari è fuori con degli amici giornalisti. Oppure è venuto un amico a trovarla e tu sei incastrato in cucina a bere una tazza di tè dopo l'altra e guardare Beavis and Butthead in inglese su MTV e a pensare di andare da qualche parte a giocare a biliardo ma fuori fa troppo freddo.

E per fortuna niente lezioni di educazione fisica per il momento.

Ti toccheranno l'anno prossimo. Per tutto l'anno. Quando l'esercito finlandese chiama a sé tutti i maschietti. Molli ragazzi fanno il servizio militare subito dopo il liceo. A me invece sembrava più sensato aspettare la fine del primo suino di università.

In Finlandia puoi scegliere: fare il militare per otto mesi o il servizio civile per un anno. Se hai forti motivazioni religiose o qualche altra scusa significativa puoi evitarli entrambi. Io non avevo questa possibilità. E non ero tanto dell'idea di fare il servizio civile.

Non era perché avessi qualcosa contro il fatto di aiutare l'umanità. Probabilmente aveva più a che fare con il timore che il servizio civile potesse essere ancora più noioso di quello militare. Non posso credere di averlo detto. Ma provate a parlare con qualcuno che abbia fatto il servizio civile e scoprirete che, se non avete già un posto che chiede la vostra assegnazione, verrete assegnati a caso in un posto privo di qualsiasi interesse. E non potevo fare obiezione di coscienza. Per quanto non avessi alcun problema a sottrarmi al mio dovere di patriota, il fatto è che io ho una coscienza: quando il gioco si fa duro, non ho grosse convinzioni contro le armi o il fatto che qualcuno venga ucciso.

Se scegli l'esercito, ti si aprono due possibilità. Puoi farti i tuoi otto mesi da soldato semplice o andare alla scuola ufficiali e fare undici mesi di leva. Io pensai che potesse essere più interessante fare l'ufficiale, nonostante i 129.600 minuti in più. E magari avrei potuto anche imparare qualcosa.

Fu così che il vostro eroe - che all'epoca pesava 60 chili -divenne sottotenente della riserva dell'Esercito Finlandese. Ero controllore di fuoco. Niente di fantascientifico. Ti danno le coordinate per l'artiglieria. Tu guardi dove ti trovi su una mappa e triangoli sul punto in cui vuoi sparare. Calcoli le coordinate e le comunichi via radio o tramite una linea telefonica che hai aiutato a stendere. Dici alle armi dove sparare.

Ricordo che prima di partire per il militare ero molto nervoso, non sapevo cosa aspettarmi. Altri avevano dei fratelli maggiori o qualcun altro con cui parlare del militare, così sapevano cosa sarebbe toccato loro. Io non avevo nessuno a dirmi cosa sarebbe successo. Be', tutti sanno in linea generale che il militare non è una cosa troppo divertente. È ciò che ti può confermare chiunque l'abbia fatto. Ma non avevo un'idea precisa di come sarebbe stato, e questo mi rendeva nervoso. È un po'come l'idea che qualcuno possa leggere questo libro.

I momenti più difficili nell'esercito erano quelli in cui mi ritrovavo a camminare in giro per la Lapponia con sulle spalle qualche tonnellata di cavi. Davvero, credo fossero davvero tonnellate di cavi. Prima di entrare alla scuola ufficiali, ti ordinano di fare cose tipo correre in giro con un enorme rotolo di cavo attorno alla pancia e due sulla schiena e devi correre magari per dieci lunghissime miglia. In altri momenti invece te ne stai lì ad aspettare che succeda qualcosa.

Oppure scii per ore e ore e poi ci si accampa. Fu lì che capii che se Dio avesse voluto che noi sciassimo ci avrebbe messo delle lunghe strisce di fibra di vetro al posto dei piedi. Anche se a dire la verità non è che io ci creda proprio, in Dio.

Poi prima di poter mangiare devi montare la tenda e accendere il fuoco. Hai freddo e sei affamato e stanco perché non dormi da due giorni. So che c'è gente che paga discrete cifre per poter partecipare a corsi di sopravvivenza di questo tipo, dicono che è roba che ti forma il carattere. Consiglierei loro di provare l'esercito finlandese.

In realtà le maratone all'aria aperta non si tenevano troppo spesso. Ma si tenevano. Ho calcolato che in quegli undici mesi nell'esercito ho passato più di 100 giorni nei boschi. La Finlandia ha un sacco di boschi: il 70 per cento del paese è coperto da foreste. Credo di averle visitate tutte.

Come ufficiale ero capocontrollore di fuoco, con quattro sottoposti. Il che significa che avrei dovuto sapere quello che facevo e farlo sembrare più complicato di quanto

fosse in realtà. Ma non era una cosa molto interessante e io non ero un grande leader. Di sicuro non ero bravo a dare gli ordini. Ero abbastanza bravo a prenderne - il trucco è non farne una questione personale - ma non sentivo che la mia missione nella vita fosse quella di fare del mio meglio.

Non lì.

Ve l'ho già detto quanto fa freddo in Lapponia?

A pensarci bene, lo odiavo il servizio militare. Ma è una di quelle cose che quando l'hai finita si trasforma subito in un'esperienza magnifica.

E poi mi ha dato qualcosa di cui parlare con quasi tutti i maschi finlandesi per il resto della mia vita. In effetti qualcuno sostiene che la ragione principale per cui il servizio militare è obbligatorio è quella di dare agli uomini finlandesi qualcosa di cui parlare davanti a un boccale di birra finché campano. Hanno tutti in comune qualcosa di orribile. Hanno odiato il servizio militare, ma dopo che è finito sono felici di parlarne.

VII.

Già che ci siamo, lasciate che vi dica qualcos'altro sulla Finlandia. Probabilmente abbiamo più renne di qualsiasi altro paese sulla faccia della terra. Abbiamo anche una discreta quantità di alcolizzati e ballerini di tango. Provate a passare un inverno in Finlandia e capirete il nostro amore per l'alcol. Per il tango non ci sono scuse, ma fortunatamente gli afecionados sono quasi tutti concentrati nei paesini e quindi non sei costretto ad averci a che fare.

Di recente una ricerca ha stabilito che i finlandesi sono gli uomini più virili d'Europa. Deve essere tutta quella carne di renna, oppure le ore passate in sauna. Nessuno sa veramente come è iniziata questa religione, ma la tradizione - perlomeno in certe zone - vuole che si costruisca prima la sauna e poi la casa. Molti condomini hanno una sauna al primo o all'ultimo piano e a ogni famiglia spetta un'ora di uso privato, tipo il giovedì dalle 19.00 alle 20.00 (i giorni della sauna sono di solito il giovedì e il venerdì). In questo modo non devono sopportare l'orrore di vedere i propri vicini di casa nudi. Una volta stavo sfogliando una guida della Finlandia in inglese che puntualizzava come i finlandesi non facciano mai sesso in sauna e che sarebbero stati sconvolti se fossero venuti a sapere che qualcuno avesse violato questa regola o che ci avesse anche solo pensato. Quando ho letto queste righe sono scoppiato a ridere e non riuscivo più a fermarmi, perché la sauna in una casa finlandese e un luogo tanto neutro che quella guida avrebbe potuto consigliare allo stesso modo ai propri lettori di non fare sesso sul pavimento della cucina. Non è niente di che, in fin dei conti. Nelle località più sperdute i bambini vengono fatti nascere nelle saune - gli unici posti in cui c'è acqua calda - e secondo alcune tradizioni è lì che si va a morire. Ma queste regole non valgono per la mia famiglia, che ha sempre avuto un approccio rilassato a tutta la faccenda.

Ci sono altri elementi che distinguono i finlandesi dagli altri membri della specie umana. Per esempio la tradizione del silenzio. La gente non parla molto. Se ne stanno lì senza dire niente. È un'altra norma che non vale per la mia famiglia, che a voler

essere generosi si potrebbe descrivere come «originalo.

I finlandesi sono stoici fino all'eccesso. Il silenzio e la determinazione ci hanno aiutato a sopravvivere alla dominazione russa, alle sanguinosissime guerre di successione e a un clima assolutamente schifoso. Ma oggi sono caratteristiche semplicemente bizzarre. Lo scrittore tedesco Bertolt Brecht visse per qualche tempo in Finlandia durante la seconda guerra mondiale ed è famosa una sua osservazione a proposito dei gestori del bar di una stazione che «restavano in silenzio in due lingue». Appena ne ebbe la possibilità, se ne andò negli Stati Uniti via Vladivostok.

Ancora oggi, se entri in un bar di una qualsiasi città finlandese - in particolare quelle più piccole - ci puoi trovale degli uomini con volti di pietra seduti da soli con lo sguardo fisso nel vuoto. La gente rispetta la privacy degli altri in Finlandia - un'altra grande cosa - per cui nessuno sognerebbe di avvicinarsi a un estraneo e mettersi a chiacchierare. I finlandesi in realtà sono persone abbastanza amichevoli. Il problema è che sono ben pochi quelli che riescono a scoprirlo.

Mi dicono che l'atmosfera è molto più conviviale nei bar finlandesi per lesbiche.

Dato che i finlandesi sono restii a conversare faccia a faccia, rappresentiamo il mercato ideale per la telefonia mobile. Abbiamo accolto questo nuovo aggeggio con un entusiasmo che non troverete in altre nazioni. Non è chiaro quale sia il paese che può vantare il maggior numero di renne pro capite - a pensarci bene il titolo potrebbe spettare alla Norvegia - ma non c'è dubbio su quale sia la nazione che detiene il record mondiale di telefoni cellulari per ogni uomo, donna e bambino. In Finlandia si dice che quando nasci ti mettono accanto il tuo bel cellulare. E i cellulari vengono usati per molti più scopi di quanto si faccia altrove. I finlandesi si mandano abitualmente messaggi di testo, o si affidano ai cellulari per barare nei compiti in classe (mandi una domanda a un amico e aspetti la riposta via SMS). Noi utilizziamo la funzione di calcolatrice di cui pochi americani sospettano anche soltanto l'esistenza. Il prossimo passo è ovvio: comporre il numero dell'estraneo che sta bevendo un cappuccino al tavolino accanto e avviare una conversazione via cellulare. A prescindere dal successo fenomenale di Nokia, i cellulari hanno cambiato la Finlandia come nient'altro dai tempi antichissimi dell'invenzione della sauna.

In effetti non è sorprendente il fatto che in Finlandia i telefoni cellulari siano stati accolti tanto bene. Si tratta di un paese che per tradizione ha adottato in modo veloce e fiducioso le nuove tecnologie. Per esempio - a differenza della maggior parte dei paesi sulla faccia della terra - la Finlandia è un posto in cui la gente d'abitudine paga le bollette e compie le proprie operazioni bancarie per via elettronica (e non con quei sistemi bancari pseudoelettronici che si trovano negli Stati Uniti). Ci sono più nodi internet pro capite in Finlandia che in qualsiasi altro paese. Alcuni attribuiscono questo amore per la tecnologia a un efficace sistema scolastico: la Finlandia ha il tasso di istruzione più alto al mondo e l'università è gratuita (il che spiega perché gli studenti ci restino per sei o sette anni. O anche - come nel mio caso - otto). Se te ne stai in un'università per così tanto tempo non puoi proprio fare a meno di imparare qualcosa. Altri sostengono che la nostra tecnofilia è stata creata dai miglioramenti delle infrastrutture dei trasporti, che facevano parte delle riparazioni pagate alla Russia dopo la guerra. Altri ancora dicono che si tratta di qualcosa che ha a che fare con l'omogeneità (a volte insopportabile) della popolazione.

Io e Linus siamo seduti al tavolo della sala da pranzo. Siamo appena tornati da un posto dove si tenevano delle corse d'auto. Tove sta mettendo via la spesa. Patricia e Daniela litigano per un libro che ho portato loro in regalo. Sposto un pinguino di peluche e un vasetto di burro d'arachidi, accendo il registratore e chiedo a Linus di parlarmi della sua infanzia.

«A dire la verità non ricordo molto della mia infanzia», dice con una voce piatta.

«Come è possibile? Non sono passati molti anni!»

«Chiedi a Tove. Io non mi ricordo bene i nomi, le facce e le cose che ho fatto. Devo sempre chiederle il nostro numero di telefono. Mi ricordo le linee generali e come sono organizzate le cose, ma non riesco mai a ricordare i particolari, nemmeno quelli della mia infanzia. Non mi ricordo come sono successe le cose o cosa pensavo quando ero piccolo.»

«Be', per esempio avevi degli amici?»

«Qualcuno. Non sono mai stato uno molto socievole. Lo sono molto di più adesso che allora.»

«Be', com'era? Voglio dire, ti ricordi di qualche volta che ti sei svegliato di domenica e sei andato da qualche parte con tua sorella e i tuoi genitori?»

«I miei erano separati all'epoca.»

«Quanti anni avevi quando si sono separati?»

«Non lo so. Tipo sei. O dieci. Non mi ricordo.»

«E Natale? Ti ricordi di qualche Natale in particolare?»

«Be', ricordo vagamente che mi mettevano in ghingheri e mi portavano a casa del mio nonno paterno, a Turku. Stessa cosa a Pasqua. Non è che ricordi molto altro.»

«E il tuo primo computer?»

«Il famoso VIC-20 comprato da mio nonno. Stava dentro una scatola.»

«Quanto era grande la scatola? Tipo quelle di un paio di doposcì?»

«Più o meno.»

«E tuo nonno? Ti ricordi qualcosa di lui?»

«Probabilmente era il mio parente più stretto, ma non... Ok. Era sovrappeso, ma non grasso. Stava perdendo i capelli. Era un tipo riservato, il tipico professore con la testa tra le nuvole. Io mi sedevo in braccio a lui e digitavo i suoi programmi.»

«Ti ricordi il suo odore?»

«No. Che razza di domanda è?»

«I nonni di solito hanno un loro odore. Acqua di colonia. Hourbon. Sigari. Il tuo che odore aveva?»

«Non saprei. Ero troppo concentrato sul computer per notarlo.»

2 La nascita di un sistema operativo [2]

I.

Certe persone ricordano il proprio passato in base alle auto che avevano o al lavoro che facevano o al posto in cui vivevano o alle ragazze con cui uscivano. I miei anni sono segnati dai computer.

Da ragazzino ho avuto solo tre computer. Il già citato Commodore VIC-20, che ereditai da mio nonno. Era uno dei primi computer «casalinghi:», il predecessore dei PC attuali. Il Commodore 64 fu una specie di fratello maggiore del VIC-20, seguito dall'Amiga, che ebbe un particolare successo in Europa. Questi computer non divennero mai veramente popolari come il PC o l'Apple II, che era già abbastanza diffuso quando iniziai a smanettare con il VIC.

A quei tempi, prima della diffusione dei PC, la maggior parte della programmazione veniva fatta in linguaggio assembly. (Non posso crederci: mi sono messo a iniziare le frasi con espressioni tipo «A quei tempi...».) I computer avevano un loro specifico sistema operativo, l'equivalente del DOS per i PC. A seconda del computer poteva essere un formato rudimentale o un po'più avanzato. Come il DOS, il sistema operativo (OS) aveva una routine per caricare programmi e un ambiente con un linguaggio di base. All'epoca non c'erano standard e una serie di aziende voleva controllare il mercato. La Commodore era una delle più conosciute.

Quando ebbi spremuto il VIC-20 al massimo delle sue possibilità iniziai a mettere via i soldi per un modello della generazione successiva. Fu un evento fondamentale della mia vita. Come ho già detto, non ricordo bene tutti gli spostamenti della mia famiglia, chi viveva con chi e un sacco di altre cose, ma la strada verso il mio secondo computer non potrei proprio dimenticarla.

Avevo messo via un po'di mance di Natale-compleanno (dato che sono nato il 28 dicembre, le due occasioni sono sempre state fuse insieme). Guadagnai qualche soldo anche con un lavoretto estivo, pulendo i parchi di Helsinki. La maggior parte dei parchi a Helsinki non sono addomesticati e ben curati, si tratta piuttosto di un incrocio tra un'area ricreativa e una foresta selvaggia. Noi dovevamo tagliare i cespugli troppo cresciuti e raccogliere i rami secchi. Era anche abbastanza interessante. Mi è sempre piaciuto stare all'aperto. A un certo punto ho fatto anche consegne in bicicletta. Pubblicità e cose del genere. In realtà non è che i lavoretti estivi mi facessero proprio impazzire. Ma in quei giorni mi ci impegnai. Facendo un po'i conti probabilmente prendevo più soldi con le borse di studio.

---

[2] Attenzione: linguaggio moderatamente geek fino a p.82

In Finlandia è normale che le persone facciano delle donazioni alle scuole, anche alle scuole elementari pubbliche. Così a partire dal quarto anno questi soldi vengono distribuiti tra gli studenti in base a ciò che aveva in mente la persona responsabile di quel fondo. Ricordo che una delle borse di studio nella mia scuola era destinata al ragazzo più popolare della propria classe. Eravamo al sesto anno e in classe si votò per chi dovesse ricevere quel premio. Non vi stupirà sapere che non fui io a vincerlo. La borsa era soltanto di 200 marchi finlandesi circa, più o meno una quarantina di dollari dell'epoca, ma sembravano un sacco di soldi da dare a un ragazzino del sesto anno per il solo fatto di essere simpatico.

Abbastanza spesso il denaro andava allo studente che riusciva meglio in una data materia o in un dato sport. E molti premi erano specifici per ciascuna scuola e sovvenzionati dal governo. In alcuni casi i fondi diminuivano nel tempo. Ricordo che uno aveva raggiunto un valore di circa un penny. Quando accadeva questo, la scuola ci aggiungeva qualcosa per renderlo un po'più significativo, ma era comunque una somma di denaro abbastanza modesta. Si trattava soprattutto di un modo per tenere viva la tradizione di elargire del denaro ogni anno. La Finlandia prende molto sul serio le proprie tradizioni accademiche, e questa è un'ottima cosa.

Così io ricevevo queste borse di studio ogni anno perché ero il Matematico. Alle superiori i premi diventavano più ricchi. Quelli più grossi erano intorno ai 500 dollari. Fu da lì che arrivò la maggior parte del denaro per il mio secondo computer: la mia paghetta mensile non sarebbe bastata. E presi anche in prestito dei soldi da mio padre.

Eravamo nel 1986 o nel 1987. Io avevo sedici o diciassette anni. Mi ero lasciato alle spalle i tempi della pallacanestro. Spesi una quantità smodata di denaro nelle ricerche sul campo per decidere quale computer comprare. I PC non erano granché all'epoca, così quando sognavo la mia nuova macchina sapevo che non sarebbe stata un PC.

Decisi per un Sinclair QL, che molti di voi potrebbero essere troppo giovani per ricordare. Eccovi la storia. Il Sinclair fu una delle prime macchine a 32 bit per uso domestico. Sir Clive Sinclair, il fondatore dell'azienda, era lo Steve Wosniak d'Inghilterra. Produceva questi computer che negli Stati Uniti venivano commercializzati con il marchio Timex. Proprio così, la stessa società che produceva gli orologi Timex importava i computer Sinclair e li vendeva con il proprio marchio. I primi esemplari venivano venduti come kit, mentre in seguito vennero commercializzati i computer pronti all'uso.

Il Sinclair aveva un sistema operativo chiamato Q-DOS. All'epoca io lo conoscevo a memoria. Era scritto specificatamente per quel computer. Aveva un Basic abbastanza avanzato per quel periodo e una grafica niente male. Una delle cose che mi piacevano di più di quel sistema operativo era il fatto che lavorasse in multitasking: potevi eseguire più programmi in contemporanea. Ma la parte in Basic non era in multitasking, per cui non potevi eseguire più di un programma in Basic per volta. Se però scrivevi i tuoi programmi in linguaggio assembly, potevi lasciare che fosse il sistema operativo a metterli in coda e temporizzarli in modo da eseguirne parecchi per volta.

Il computer conteneva un chip 68008 a 8 megahertz, che era la seconda (e più

economica) versione del chip Motorola 68000. All’interno i chip 68000 della prima generazione erano a 32 bit, ma all’esterno l’interfaccia con tutto ciò che stava fuori dalla CPU (Central Processing Unit, l’unità di calcolo centrale) - come per esempio la memoria o le periferiche hardware - era a 16 bit. Dato che poteva caricare dalla memoria solo 16 bit alla volta, le operazioni a 16 bit erano spesso più veloci di quelle a 32 bit. Era un’architettura molto popolare, utilizzata ancora oggi in molte periferiche interne o sulle automobili. Non è lo stesso chip, ma è basato sulla stessa architettura.

Il chip 68008, la versione montata sul mio computer, usava 8 bit - e non 16 - per interfacciare il mondo esterno alla CPU. Ma anche se interagiva con il mondo esterno a 8 bit alla volta, all’interno era a 32 bit. Questo lo rendeva per molti aspetti più piacevole da programmare.

Aveva una memoria di 128 kilobyte - non megabyte - che all’epoca per una macchina casalinga non era affatto poca. Il VIC-20 che avrebbe sostituito aveva solo $3\frac{1}{2}$ kilobyte di memoria. E dato che si trattava di una macchina a 32 bit, poteva accedere a tutta la memoria senza alcun problema, una cosa del tutto nuova per l’epoca. Questa era la ragione principale per cui volevo comprare quel computer. La tecnologia era interessante e adoravo la CPU.

Speravo di avere il computer con un certo sconto comprandolo in un negozio di cui un mio amico conosceva i proprietari. Ma avrei dovuto aspettare così tanto tempo per averlo che me ne andai alla Akademiska Bokhandeln, la più grande libreria di Helsinki, che aveva un reparto informatica. Lo comprai lì.

Il computer costava più o meno 2000 dollari. C’era questa specie di regola per cui i computer entry-level costavano sempre 2000 dollari. È solo da un paio d’anni a questa parte che questa regola è cambiata. Oggi ti puoi comprare un PC nuovo per 500 dollari. E come per le auto. Nessuno fa auto che costino meno di 10.000 dollari. Arrivati a un cerio punto, non ne vale più la pena. Certo, le aziende automobilistiche potrebbero costruire un’auto da vendere per 7000 dollari, ma secondo il loro ragionamento uno che può permettersi di spendere 7000 dollari per un’auto sarà ben felice di cacciarne 10.000 per averne una con più accessori tipo aria condizionata. Se confrontate le auto entry-level di quest’anno con quelle di quindici anni fa, il prezzo è più o meno lo stesso. In effetti, una volta fatti i conti con l’inflazione, oggi forse costano un po’di meno. Ma sono molto migliori.

Anche per i computer era così. Quando erano una cosa che non tutti compravano, c’era una soglia iniziale che si aggirava intorno ai 2000 dollari. Se il computer di prezzo più basso costa molto di più, l’azienda non ne potrà vendere molti. Ma la loro produzione era abbastanza costosa e non avrebbe avuto senso per l’azienda abbassare sensibilmente il prezzo. I clienti avrebbero sicuramente pagato 200 dollari in più per una macchina leggermente migliore. Da quando i produttori non hanno più potuto competere solo sulle caratteristiche tecniche, hanno dovuto iniziare a farlo sui prezzi.

Devo ammetterlo: nel 1987 uno dei motivi per comprare il QL era che aveva un look figo.

Era nero opaco, con una tastiera nera. Molto angoloso. Non era una macchinetta tondeggiante da bravi ragazzi. Si dava un contegno in qualche modo estremo. La tastiera era spessa poco più di due centimetri perché faceva parte della stessa unità del

computer. La maggior parte dei computer casalinghi erano progettati a questo modo. Sulla destra della tastiera, dove in seguito avrebbe trovato posto il tastierino numerico, c'erano due slot per il rivoluzionario microdrive Sinclair, un nastro interminabile utilizzato solo sulle macchine Sinclair. Funzionava ed era organizzato come un lettore per dischetti. Dato che era un lungo nastro, bastava avvolgerlo per raggiungere il punto desiderato. Alla fine si dimostrò un'idea mediocre, perché non era affidabile quanto un lettore per dischetti.

Così spesi quasi 2000 dollari per il Sinclair QL. Cosa me ne feci? Un programma dopo l'altro. Cercavo sempre qualcosa di interessante da fare. Avevo un interprete e un compilatore di linguaggio Forth e mi ci trastullavo. Il Forth era uno strano linguaggio che oggi non usa più nessuno. Si trattava di un linguaggio divertente, una cosa di nicchia, e negli anni Ottanta era abbastanza diffuso per diverse funzioni, ma non divenne mai molto popolare perché era difficile da seguire per i non addetti. In effetti era fondamentalmente inutile.

Io mi scrivevo i tool di programmazione da solo. Una delle prime cose che comprai per il nuovo computer fu una console di espansione con una scheda eprom (Elettrically Erasable and Programmable Read Only Memory, memoria di sola lettura cancellabile e programmabile elettronicamente). È un tipo di memoria su cui puoi scrivere con dei moduli speciali, che non si cancella quando spegni la macchina. In quel modo potevo tenere i tool sempre a mia disposizione senza doverli caricare nella RAM (Random Access Memory) e risparmiando la preziosa RAM per i programmi.

Com'è che iniziai a interessarmi di sistemi operativi? Mi comprai un controller per dischetti, così non avrei dovuto usare i microdrive, ma il driver che accompagnava il controller era pessimo e finii per scrivermene uno da solo. Scrivendo questo driver scoprii dei bachi nel sistema operativo - o quantomeno delle discrepanze tra quello che secondo la documentazione il sistema operativo avrebbe dovuto fare e quello che di fatto faceva. Li scoprii perché parte del codice che avevo scritto non funzionava.

Il mio codice è sempre... ehm... perfetto. Per cui sapevo che doveva esserci qualcos'altro che non andava e mi misi a disassemblare il sistema operativo.

Ti puoi comprare dei libri che contengono dei listati parziali del sistema operativo. Aiuta. Hai anche bisogno di un disassembler, un tool che prende il linguaggio macchina e lo traduce in linguaggio assembly. È importante, perché quando hai solo una versione in linguaggio macchina è difficile seguire le istruzioni. Scopri che un'istruzione punta a un indirizzo numerico, il che rende molto difficile leggere Il tutto. Un buon disassembler dà dei nomi ai numeri e ti permette anche di specificarne di tuoi. Può essere usato anche per aiutarti a identificare particolari sequenze di istruzioni. Io avevo il mio disassembler che potevo usare per creare dei listati abbastanza decenti. Quando qualcosa non funzionava, potevo entrare e dirgli di trovare il listato a partire da un dato punto e vedere così tutto ciò che avrebbe fatto il sistema operativo. A volte usavo il disassembler non perché c'era un elemento che non funzionava, ma per capire come funzionava.

Una delle cose che detestavo nel QL era il fatto che avesse un sistema operativo di sola lettura. Non potevi cambiate niente. C'erano dei punti in cui potevi inserire il tuo codice per controllare determinate funzioni, ma si trovavano tutti in posizioni predefinite. È molto meglio poter sostituire completamente il tuo sistema operativo.

Fare un sistema operativo nella rom (Read-Only Memory, memoria di sola lettura) è una pessima idea.

Nonostante quanto ho detto sull'amore della Finlandia per la tecnologia, il Sinclair QL non stava avendo un grande successo nel settimo paese più grande d'Europa. Poiché Il mercato era piccolo, quando volevi comprare degli aggiornamenti per quella macchina iconoclasta e all'avanguardia dovevi farlo per posta, dall'Inghilterra. Il che voleva dire scartabellare un sacco di cataloghi finché non trovavi qualcuno che vendeva quello che stavi cercando. Poi dovevi mettere insieme degli assegni postali e aspettare per settimane la consegna (erano ancora lontani i tempi di Amazon.com e delle carte di credito). Fu quello che dovetti fare anch'io quando decisi di espandere la mia RAM da 128 a 640 kilobyte. E la stessa cosa accadde quando comprai un nuovo assembler per tradurre il linguaggio assembly in linguaggio macchina (gli uno e gli zero) e un editor, che è fondamentalmente un elaboratore di testi per la programmazione.

Sia il nuovo assembler sia l'editor funzionavano bene, ma erano su microdrive e non li potevo mettere sulla E-PROM. Così mi misi a scrivere un editor e un assembler e li utilizzai per tutti i miei programmi. Li scrissi tutti e due in linguaggio assembly, che per gli standard di oggi è una cosa incredibilmente stupida. È complicato e porta via un sacco di tempo: credo che per risolvere un problema in linguaggio assembly ci voglia cento volte più tempo di quanto ne serva per risolverlo per esempio in C, un linguaggio che era già disponibile all'epoca.

Aggiunsi alcuni comandi all'interprete di base integrato nella macchina, così se volevo editare qualcosa dovevo solo eseguire automaticamente il mio editor e me lo ritrovavo immediatamente lì. Il mio editor era più veloce di quello integrato nella macchina. Ero particolarmente fiero della velocità con cui potevo visualizzare i caratteri a schermo. Di norma, con una macchina del genere, ci voleva così tanto tempo per riempire lo schermo di caratteri che riuscivi a vedere lo scorrimento del testo. E io ero molto felice del fatto che con il mio editor visualizzavi il testo tanto velocemente che quando lo facevi scorrere verso il basso si creava una specie di effetto flou. Era una cosa importante, per me. Quel miglioramento dava alla macchina un'aria più scattante e io sapevo di avere fatto un sacco di lavoro per farla funzionare a quel modo.

In quel periodo non c'erano molte persone di mia conoscenza appassionate di computer come lo ero io. A scuola c'era un club informatico, ma non ci passavo molto tempo. Era una cosa per ragazzini che volevano sapere qualcosa sui computer. C'erano più o meno 250 studenti nel mio liceo e non credo che qualcun altro abbia usato un computer prima dei dieci anni.

Una delle cose che mi piaceva fare con il mio Sinclair QL era clonare i giochi. Scrivevo dei cloni dei miei giochi preferiti per il VIC-20 e a volte aggiungevo delle cose. Ma perlopiù non erano meglio degli originali: una macchina migliore, non un concetto migliore.

Il mio gioco preferito era probabilmente Asteroids, ma non sono mai riuscito a farne un buon clone. Il motivo era che all'epoca tutte le versioni da bar di Asteroids erano fatte con una grafica vettoriale. Invece di avere una grafica basata sui puntini - i pixel - ne avevano una basata esattamente sul modo di funzionare del tubo catodico

(CRT), ovvero con degli elettroni sparati da un cannone a elettrodi dietro il CRT e deviati tramite dei magneti. In quel modo ottenevano una grafica con risoluzione maggiore, ma non era facile da riprodurre. Si poteva farne un clone, ma non avrebbe avuto l'aspetto dell'Asteroids originale se lo scrivevi su un computer che non aveva particolari capacità grafiche.

Ricordo di avere fatto un clone di Pac Man in linguaggio assembly. Il primo passaggio consiste nel cercare di ricordare l'aspetto dei personaggi di Pac Man. Poi cerchi di disegnarli su una griglia di carta 16 per 16, a colori. E se sei un tipo artistico puoi anche fare un buon lavoro. Ma se sei Mister Antiartista, come me, finisci per creare delle specie di cugini mostruosi di Pac Man.

Ok, non era un granché come clone. Ma io ne ero molto fiero. Ci si poteva giocare per davvero e lo mandai a una rivista che pubblicava codici di programmi informatici. Avevo venduto altri programmi alle riviste e pensavo che sarebbe successo anche questa volta.

Niente.

Tanto per cominciare il programma era scritto in linguaggio assembly. Il che significava che se facevi il minimo (e intendo davvero minimo) errore nel copiarlo dalla rivista, non avrebbe funzionato.

Scrissi anche dei giochi inventati da me. Ma ci vuole la mentalità giusta per creare dei giochi. Dato che i giochi hanno bisogno di prestazioni notevoli, devi andare veramente a fondo nell'hardware del computer. Io lo sapevo fare, ma non avevo la mentalità giusta per i giochi. La grandezza di un gioco non dipende di solito da quanto è veloce o buona la grafica. Ci deve essere qualcosa che ti fa venire voglia di giocarci, qualcosa che ti aggancia. È come per i film. Gli effetti speciali vanno bene, ma ci vuole anche una trama. E i miei giochi non avevano mai una trama. Un gioco deve avere una progressione, un'idea. Spesso la progressione sta semplicemente nel fatto che il gioco diventa più veloce. Pac Man funziona così. A volte cambia il labirinto o i mostri diventano più bravi a inseguirti.

Una delle cose che mi interessavano in Pac Man era il problema di creare una grafica priva di sfarfallii. È un problema abbastanza comune nei vecchi giochi per computer, perché senza un hardware speciale i personaggi si limitano a sfarfallare. Per spostare i personaggi da una parte all'altra si toglie la vecchia copia e se ne scrive una nuova. Se non hai il tempo giusto, chi gioca può vedere il momento in cui non c'è alcuna copia. Di qui lo sfarfallio. È un problema che si può aggirare in molti modi. Puoi disegnare prima il nuovo personaggi e poi cancellare quello vecchio, ma devi stare attento a non togliere la parte del vecchio personaggio occlusa da quello nuovo. Invece di vedere un irritante sfarfallio, ottieni un buon effetto: a volte vedi l'ombra del vecchio personaggio sullo schermo. Il cervello interpreta in modo positivo questa cosa. Non vede uno sfarfallio, ma l'impressione di un movimento. Il problema di questa soluzione è che è abbastanza costosa e porta via molto tempo.

C'è un motivo per cui i giochi sono sempre all'avanguardia e per cui sono spesso il primo tipo di programmi creati da un programmatore. In parte ha a che fare con il fatto che alcuni dei migliori programmatori là fuori sono dei quindicenni che giocano nelle loro stanzette (lo pensavo sedici anni fa e ancora oggi sospetto sia vero). Ma c'è un'altra ragione per cui i giochi sono all'avanguardia, ed è perché tendono a spingere

l’hardware fino al limite.

Se pensate ai computer di oggi, di solito sono abbastanza veloci per farci qualsiasi cosa. Ma per mettere alla prova i limiti dell’hardware si usano i giochi d’azione, tipo quelli in 3d che oggi sono tanto popolari. Fondamentalmente i giochi sono una delle poche cose (per quanto riguarda i computer) in cui puoi capire se le cose non stanno accadendo in tempo reale. Nell’elaborazione di testi un ritardo di un secondo qui e un secondo là non crea alcun problema. In un gioco invece inizi a notare un ritardo quando supera il decimo di secondo. I giochi una volta erano abbastanza semplici. Oggi la programmazione è una parte relativamente piccola in un gioco qualsiasi. C’è la musica, la storia. Se lo confronti alla realizzazione di un film, il lavoro di programmazione corrisponde a quello degli operatori.

Tenni il Sinclair QL per tre anni. Mi portò dal liceo all’Università di Helsinki all’Esercito Finlandese. Non era male, ma a quel punto eravamo decisamente pronti a prendere ciascuno la propria strada. Nell’ultimo anno avevo scoperto i suoi limiti. Il 68008 era una discreta CPU, ma leggevo dappertutto della nuova generazione, il 68020, e di nuove virtù come la gestione della memoria e il paging. Questi nuovi computer potevano fare delle cose davvero importanti se stai lavorando su roba di basso livello.

Ciò che mi irritava nel Sinclair QL era che anche se il sistema operativo era in multitasking, la macchina si poteva bloccare in ogni momento perché non c’era una protezione della memoria. Un’applicazione che decideva di fare qualcosa di sbagliato poteva imballarti il computer.

Il Sinclair QL fu l’ultimo parto di Sir Clive Sinclair nella progettazione e produzione di computer. Un motivo tra tanti: non aveva avuto successo commerciale. Aveva una tecnologia interessante, ma l’azienda aveva problemi di produzione, di controllo qualità e di accoglienza da parte della stampa. E poi il mercato stava iniziando a diventare più competitivo.

I tardi Ottanta erano gli anni in cui potevi iniziare a immaginare che, ebbene sì, forse un giorno anche i commessi dei supermercati avrebbero avuto un computer, anche se da usare solo come elaboratore di testi. E tutti gli indizi puntavano verso il PC. Sì, i PC IBM originali avevano iniziato a invadere gli scaffali dei negozi e ad avere successo nonostante i numerosi difetti tecnici. Queste ubique creature beige avevano il marchio d’approvazione IBM, in fondo, e questo voleva dire molto. Un altro elemento d’attrazione: le periferiche erano standard ed era facile procurarsele.

Io leggevo di tutte queste nuove CPU che potevano fare ciò che io volevo. Era chiaro che il 68020, che sembrava interessante, non aveva un futuro. Avrei potuto considerare l’ipotesi di comprare un aggiornamento per la CPU del QL. All’epoca voleva dire sostanzialmente ricostruire la macchina. E il sistema operativo in ogni caso non sapeva nulla di gestione della memoria, per cui avrei dovuto scriverne una mia versione. Pensavo qualcosa del tipo: Mmmm... Certo che sarebbe un bel passo. E prendere una nuova CPU mi costerebbe un sacco di soldi.

Poi c’era anche il problema sempre più spiacevole di comprare le periferiche per il computer. Non c’era un catalogo tipo Postalmarket per il Sinclair QL, una cosa per cui potevi prendere il telefono e ordinare un po’di memoria. La faccenda degli ordini postali dall’Inghilterra stava diventando vecchia. (Non mi interessava non trovare

software preconfezionato, perché quello me lo potevo scrivere da solo.)

Ma questa rottura di scatole ebbe un effetto collaterale positivo. Quando stavo pensando di disfarmi del mio computer, decisi di vendere tutte le periferiche: il vero disco fisso che avevo comprato perché non ne potevo più del microdrive e l'espansione RAM. Ma non è che la gente facesse la fila per comprare questo genere di cose, per cui dovetti pubblicare un annuncio su una rivista di informatica e incrociare le dita. Fu così che incontrai il mio amico Jouko Vierumaki, che era probabilmente l'unica altra persona in tutta la Finlandia a possedere un Sinclair QL. Rispose alla mia inserzione, prese un treno da Lahti e comprò un po'delle mie periferiche. E poi mi fece scoprire il biliardo.

II.

Durante il mio primo anno di università, il Sinclair QL trovò posto su una scrivania davanti alla finestra della mia stanza al primo piano su Petersgatan, ma non feci grandi passi in avanti con la programmazione. In parte perché volevo concentrarmi sugli studi. Ma anche perché mi ritrovai senza un progetto da realizzare sul mio computer. E senza un progetto ti manca pure l'entusiasmo. Cerchi di pensare a qualcosa che ti possa motivare.

Mi sembrava il momento giusto per fare il militare, che prima o poi tanto mi sarebbe toccato. Avevo diciannove anni, ero irritato dai limiti del mio computer e non avevo progetti informatici interessanti. Presi un treno per la Lapponia.

Ho già spiegato quanto fossi impreparato, tra le altre cose, alle fatiche fisiche del servizio militare. Così dopo undici mesi di ginnastica armata mi sentii del tutto giustificato a passare i decenni che mi restavano da vivere nella più totale inattività. Il mio unico esercizio fisico consisteva nel digitare il codice su una tastiera o nello stringere tra le dita un boccale di birra. (In effetti la mia prima attività più o meno sportiva dopo la fine del servizio militare ebbe luogo almeno dieci anni dopo il congedo, quando David mi costrinse ad affrontare su una tavola le onde assassine di Half Moon Bay. Sono quasi annegato e ho avuto le gambe a pezzi per giorni e giorni.)

Il servizio militare finì il 7 maggio 1990. Tove potrebbe dirvi - a ragione - che fatico a ricordare la data del nostro anniversario, ma non potrei mai dimenticare quella del mio congedo.

La prima cosa che volevo fare era prendere un gatto.

Avevo un amico la cui gatta aveva partorito qualche settimana prima, per cui comprai l'ultimo gattino rimasto, che era bianco, maschio e bellissimo. Dato che aveva passato le sue prime settimane all'aperto si adattò perfettamente a vivere sia dentro sia fuori dall'appartamento di mia madre. Lo chiamai Randi, diminutivo di Mithrandir, il mago bianco del Signore degli anelli. Oggi ha undici anni e, come il suo proprietario, si è adattato perfettamente allo stile di vita californiano.

No, non penso di avere fatto alcunché di produttivo nel corso di quell'estate. I corsi del mio secondo anno di università sarebbero iniziati solo in autunno. Il mio

computer non era in gran forma. Così me ne giravo per casa con addosso un accappatoio logoro, giocavo con Randi e ogni tanto mi trovavo con gli amici in modo che potessero prendermi in giro per i miei tentativi di giocare a bowling o a biliardo. Ok, lo ammetto, un po'di tempo lo passavo a sognare a occhi aperti il mio prossimo computer.

Mi trovavo di fronte a un tipico dilemma da geek. Come tutti i puristi cresciuti con un chip 68008, snobbavo i PC. Ma quando nel 1986 uscì il processore 386, i PC iniziarono a diventare più attraenti. Potevano fare tutto ciò che faceva un 68020, e nel 1990 la produzione di massa e l'introduzione di cloni poco costosi li resero molto più convenienti. Ero molto attento alla questione soldi, perché non ne avevo. Così decisi: è questa la macchina che voglio. E dato che i PC erano in piena espansione, gli aggiornamenti e le periferiche sarebbero stati facili da trovare. Soprattutto per quanto riguardava l'hardware, volevo avere qualcosa di standardizzato.

Decisi di salire a bordo, di saltare il fosso. Sarebbe stato divertente avere una nuova CPU. Fu a quel punto che iniziai a vendere i pezzi del mio Sinclair QL.

Oggi tutti hanno un libro che ha cambiato loro la vita. La Bibbia. Il capitale. I giovedì con Maury. Tutto quello che c'è da sapere l'ho imparato all'asilo. Quello che vi pare. Io spero sinceramente che, avendo letto la prefazione e la mia teoria sul Senso della Vita, deciderete che questo è il libro che cambierà la vostra vita. Quello che si prese la mia e la proiettò verso nuove altezze fu Sistemi operativi: progettazione e implementazione di Andrew S. Tanenbaum.

Mi ero già iscritto ai corsi autunnali e quello che aspettavo con maggiore ansia era il corso sulla programmazione in C e sul sistema operativo Unix. Quell'estate, per prepararmi al corso, comprai il libro che ho appena citato nella speranza di portarmi avanti col lavoro. Nel libro Andrew Tanenbaum, un professore universitario di Amsterdam, parlava di Minix, un programma didattico che aveva scritto in Unix. Minix è anche un piccolo clone di Unix. Subito dopo avere letto l'introduzione e avere scoperto la filosofia di Unix e quello che quel sistema operativo potente, pulito e bellissimo era in grado di fare, decisi di prendermi una macchina su cui far girare Unix. Avrei installato Minix, l'unica versione ragionevolmente utile che potessi trovare.

A mano a mano che leggevo e iniziavo a capire Unix, sentivo l'entusiasmo che mi cresceva dentro. E sinceramente non se n'è mai più andato. Spero che anche voi possiate dire lo stesso di qualche cosa.

III.

L'anno accademico che iniziò nell'autunno del 1990 fu il primo in cui l'Università di Helsinki adottò Unix, il potente sistema operativo nato ai Bell Labs della AT&T nei tardi anni Sessanta e cresciuto in giro per il mondo. Durante il mio primo anno di studi, avevamo un VMS che girava su VAX. Il VMS era un sistema operativo orribile, decisamente non si trattava di un ambiente che ti faceva dire «Cavoli, quanto vorrei averlo anche a casa». Era più probabile che pensassi qualcosa del tipo:

«Mmmmm. Ma come l'hanno fatto questo *coso?*» Era difficile da usare. Non aveva molti tool. Non era adatto a entrare facilmente in internet, che girava su Unix. Non era facile nemmeno capire quanto fosse grande un file. Devo ammettere che il VMS era ottimo per alcune operazioni, tipo i database. Ma non è il tipo di sistema operativo per cui puoi perdere la testa.

L'università aveva capito che era ora di fare un passo in avanti. Buona parte del mondo accademico era sempre più innamorata di Unix, così l'università comprò un MicroVAX su cui girava Ultrix, la versione di Unix della Digital Equipment Corporation. Era un modo per mettere un piede nel mondo di Unix.

Io non vedevo l'ora di lavorare con Unix per sperimentare le cose che imparavo dal libro di Andrew Tanenbaum, ero eccitatissimo all'idea delle cose che avrei potuto esplorare se avessi avuto un PC 386. Ma non avevo alcuna possibilità di mettere insieme i 18.000 marchi finlandesi necessari per comprarne uno. Sapevo che una volta cominciato il semestre autunnale avrei potuto usare il mio Sinclair QL per entrare nel nuovo computer Unix dell'università finché non mi fossi potuto permettere un PC su cui far girare un Unix tutto mio.

Così quell'estate feci due cose. Niente. E leggere le 719 pagine di *Sistemi operativi: progettazione e implementazione.* Quel manuale in brossura rossa ormai viveva sul mio letto.

L'Università di Helsinki acquistò una licenza da sedici utenti per il MicroVAX. Il che voleva dire che l'accesso al corso «C e Unix» sarebbe stato limitato a trentadue studenti (immagino che l'idea fosse che sedici l'avrebbero usato di giorno e sedici di notte). Come tutti noi, anche l'insegnante era un novizio di Unix. Lo ammise subito, per cui la cosa non era un problema. Ma leggeva il libro di testo con un solo capitolo di vantaggio sugli studenti, mentre questi a volte saltavano in avanti di due o tre capitoli. Così divenne una specie di gioco in cui la gente cercava di mettere all'angolo il professore facendogli delle domande che riguardavano cose che avremmo imparato solo tre capitoli più tardi, tanto per vedere se li aveva già letti.

Eravamo tutti bambini sperduti nella foresta di Unix, con un corso che si costruiva strada facendo. Ma la cosa che risultò chiara da questo corso fu che esisteva una filosofia unica dietro Unix. Lo capivi dopo la prima ora di lezione. Il resto erano dettagli.

La cosa speciale di Unix è l'insieme di idee che porta avanti. È un sistema operativo pulito e bello. Evita le eccezioni. Unix è basato sull'idea di processo: un processo è qualcosa che fa qualcosa. Ecco un semplice esempio. In Unix comando di *shell,* ovvero quello che digiti per entrare nel sistema operativo, non è integrato nel sistema operativo, come accade per il DOS. È semplicemente un'applicazione come tutte le altre. Solo che quest'applicazione legge dalla tua tastiera e scrive sul monitor. Tutto ciò che in Unix fa qualcosa è un processo. E poi ci sono i file.

Era la semplicità del design di Unix ad affascinare me e li maggior parte delle persone (be', quantomeno la maggior parte dei *geek).* Quasi tutto quello che fai in Unix viene fatto con soltanto sei operazioni di base (che si chiamano «chiamate di sistema» perché sono delle chiamate che fai al sistema operativo per fargli fare delle cose per te). E puoi costruire più o meno qualsiasi cosa a partire da quelle sei chiamate di sistema.

E poi c'è la nozione di *fork,* che è una delle operazioni fondamentali di Unix. Quando un processo fa *una fork,* crea una copia completa di sé stesso. In questo modo ottieni due copie identiche. La seconda copia spesso finisce per *eseguire* un altro processo, sostituendosi con un nuovo programma. E questa è la seconda operazione di base. Poi ci sono quattro altre chiamate di sistema fondamentali: *open,* cl*ose, read* e *write,* tutte progettate per accedere ai file. Oneste sei chiamate di sistema creano le semplici operazioni da cui è composto Unix.

Certo, ci sono tonnellate di altre chiamate di sistema per risolvere ogni dettaglio. Ma una volta che hai capito le sei fondamentali, hai capito Unix. Perché una delle cose belle di Unix è come ti fa capire che non hai bisogno di interfacce complesse per costruire qualcosa di complesso. Puoi arrivare a qualsiasi livello di complessità partendo dall'interazione di elementi semplici. Quello che devi fare è creare dei canali di comunicazione (chiamati *pipe* in gergo Unix) tra processi semplici per risolvere problemi complessi.

Un brutto sistema è quello in cui ci sono interfacce speciali per qualsiasi cosa tu debba fare. Unix è il contrario. Ti fornisce i mattoni con cui puoi costruire qualsiasi cosa. Un design pulito non è nient'altro che questo.

È esattamente come per le lingue. L'inglese ha ventisei lettere e puoi costruire tutto a partire da quelle lettere. All'estremità opposta c'è il cinese, in cui hai una lettera per ogni cosa che puoi pensare. In cinese parti dalla complessità e la complessità la puoi combinare solo entro certi limiti. E un po'come l'approccio del VMS, avere elementi complessi che hanno significati interessanti ma non possono essere usati in modo diverso. È anche l'approccio di Windows.

Unix, invece, è basato sulla filosofia «piccolo è bello. Ha un piccolo corpo di mattoni di base che possono essere combinati in un'infinita complessità d'espressione.

Anche la fisica, tra l'altro, funziona a questo modo. Cerchi di trovare le leggi fondamentali, che dovrebbero essere abbastanza semplici. La complessità nasce dalle molte e incredibili interazioni che si ottengono da queste leggi semplici, non da una complessità intrinseca delle leggi stesse.

La semplicità di Unix non era un caso. Unix, con la sua idea dei mattoni di base, è stato creato e scritto con grande fatica da Dennis Richie e Ken Thompson ai Bell Labs della AT&T. E non dovete assolutamente confondere la semplicità con la facilità. Ci vogliono senso del design e buongusto per essere semplici.

Per tornare all'esempio delle lingue: le scritture ideografiche come quella cinese e i geroglifici tendono a comparire per prime e a sembrare più «facili», mentre la scrittura «a mattoni» richiede un approccio più astratto. Allo stesso modo, non dovreste confondere la semplicità di Unix con una mancanza di sofisticazione. In realtà è vero il contrario.

Il che non vuol dire che le motivazioni iniziali alla base Unix fossero molto sofisticate. Come tante altre cose in informatica, era una questione di giochi. C'era qualcuno e voleva giocare su un PDP-11. Fu per questo che si iniziò a sviluppare Unix, con un progetto personale di Dennis e Ken per giocare a *Space Wars.* E dato che il sistema operativo non era considerato un progetto serio, l'AT&T non lo ritenne un'operazione commerciale. In effetti l'AT&T era un monopolio regolamentato e in

ogni caso una delle cose che non potevano fare era vendere computer. Per cui le persone che crearono Unix lo resero disponibile abbastanza liberamente insieme alle licenze delle sorgenti, soprattutto alle università. Non era un grosso affare.

Tutto ciò portò Unix a diventare un grande progetto nei circoli accademici. Con la sentenza del 1984, quando alla AT&T venne finalmente consentito di entrare nel settore informatico, gli studiosi di computer nelle università - in particolare all'University of California-Berkeley - lavoravano su Linux (apportandovi dei miglioramenti) da diversi anni sotto la direzione di persone come Bill Joy e Marshall Kirk McKusik. E non tutti si preoccupavano troppo di documentare ciò che facevano.

Agli inizi degli anni Novanta Unix era diventato il sistema operativo numero uno per tutti i supercomputer e i server. Era un grande business. Il problema era però che c'era ormai tutta una serie di versioni diverse del sistema operativo. Alcune erano derivate dai confini più controllati si codice base dell'AT&T (il cosiddetto «System V»), mentre altre erano derivate dal codice base dell'University of California-Berkeley, il BSD (Berkeley Software Distribution). Altre ancora erano un misto di entrambi i codici.

Vale la pena di citare in particolare una derivazione del BSD. Si tratta del progetto 385BSD realizzato da Bill Jolitz sulla base del codice BSD e distribuito via internet. In seguito sarebbe stato frammentato e come origine dei codici liberi di tipo BSD (NetBSD, FreeBSD e OpenBSD) avrebbe attirato molta attenzione sulla comunità Unix.

Fu per questo che la AT&T si svegliò e fece causa all'University of California-Berkeley. Il codice originale era dell'AT&T, ma la maggior parte del lavoro successivo era stato fatto a Berkeley. Gli amministratori della University of California sostenevano di avere il diritto di distribuire o vendere per un prezzo simbolico la propria versione di Unix. E dimostrarono di avere fatto così tanto lavoro che il materiale messo a disposizione da AT&T era stato sostanzialmente riscritto. La causa finì con un patteggiamento quando la Novell acquistò Unix dall'AT&T. In pratica si dovettero eliminare alcune parti del sistema da ciò che l'AT&T aveva reso disponibile.

Nel frattempo tutte queste dispute legali erano servite a dare all'ultimo venuto il tempo per maturare e diffondersi. In pratica avevano concesso a Linux la possibilità di conquistare il mercato. Ma sto correndo troppo.

Dato che sono già partito con le digressioni, mi piacerebbe spiegarvi una cosa. Unix ha la reputazione di una specie di magnete per le frange più eccentriche dell'informatica. È una reputazione che non vale nemmeno la pena di discutere. É vera.

Sinceramente, c'è *un sacco* di gente abbastanza folle in Unix. Non di quei folli che mandano lettere minatorie. Nemmeno di quelli che avvelenano il cane del vicino. É semplicemente gente con uno stile di vita *molto* alternativo.

Ricordatevi che la maggior parte dell'attività iniziale di Unix ha avuto luogo alla fine degli anni Sessanta e nei primi anni Settanta, mentre io dormivo nel cestino della biancheria nelle case dei miei nonni. Era gente tipo flower-power. Ma flower-power *tecnologico.* Molta della filosofia «Unix deve essere libero» ha più a che fare con lo spirito dell'epoca che con il sistema operativo. Era un periodo di grande idealismo.

Rivoluzione. Libertà dalle autorità. Amore libero (che io mi sono perso, anche se in ogni caso è probabile che non avrei saputo bene cosa farmene). E la relativa apertura di Unix, anche se era dovuta soprattutto alla momentanea mancanza di interesse commerciale, attraeva questa gente come falene.

La prima volta che mi venne presentato questo aspetto di Unix fu probabilmente intorno al 1991, quando Lars Wirzenius mi trascinò a una conferenza del Politecnico di Helsinki (che, come tutti sanno, non si trova veramente a Helsinki ma dall'altra parte del confine, a Espio. Però si chiama così perché chi l'ha fondato voleva che il nome ricordasse la scintillante capitale finlandese). L'oratore era Richard Stallman.

Richard Stallman è il Dio del Free Software.[3] Aveva iniziato a lavorare a un'alternativa a Unix nel 1984 e l'aveva chiamata sistema GNU. GNU sta per «GNU is Not Unix» *(GNU Non* è *Unix):* si tratta di uno dei molti acronimi autoreferenziali in cui una delle lettere sta per l'acronimo stesso, una specie di scherzetto tra informatici che nessun altro capisce mai. Noi *geek* siamo uno spasso, eh?

Ma soprattutto RMS (come preferisce essere chiamato Richard) ha scritto il *Free Software Manifesto* e la licenza di copyright per il Free Software, la GPL (General Public License, *licenza pubblica generale).* Fondamentalmente è stato un pioniere dell'idea della libera disponibilità del codice sorgente come qualcosa di intenzionale, e non solo un caso come è accaduto per l'originale sviluppo aperto di Unix.

Devo ammettere che io non mi interessavo molto delle questioni sociopolitiche che erano - e sono ancora - tanto care a RMS. Non sapevo nemmeno che esistesse la Free Software Foundation (che lui aveva creato) e quali fossero i suoi obiettivi. A giudicare dal fatto che non ricordo molto di quella conferenza del 1991, probabilmente in quella fase non ebbe un impatto fortissimo sulla mia vita. Mi interessava la tecnologia, non la politica. Di quella ne avevo abbastanza a casa. Ma Lars era un ideologo e io rimasi seduto ad ascoltarlo.

In Richard vidi per la prima volta nella mia vita lo stereotipo barbuto e capellone dell'hacker. A Helsinki non è che ce ne siano molti.

Forse non ricevetti un'illuminazione, ma credo che qualcosa di quel discorso mi dovette entrare dentro. Dopo tutto più tardi avrei usato la GPL per Linux. Ma ecco che ci risiamo, mi sono messo di nuovo a correre troppo.

IV.

2 gennaio 1991. Era il primo giorno in cui i negozi erano aperti dopo Natale e dopo il mio ventunesimo compleanno, i due giorni di maggiori entrate economiche dell'intero mio calendario.

Con i miei soldi di Natale-compleanno in mano presi la grande decisione economica di acquistare un computer che mi sarebbe costato 18.000 marchi finlandesi, ovvero più o meno 3500 dollari. Non avevo tutti quei soldi, per cui l'idea

---

[3] Non traduciamo Free Software in italiano per diversi motivi. In primo luogo è una locuzione ormai estremamente diffusa anche nel nostro paese. In secondo luogo il termine free significa sia libero sia gratuito e soprattutto in questo caso è preferibile conservare l'ambiguità dell'espressione originale [n.d.t]

era quella di pagare un terzo in contanti e il resto a rate. In realtà il computer costava 15.000 marchi. Il resto erano interessi sulle rate che avrei pagato per i tre anni successivi.

Comprai il computer in un negozietto a gestione familiare. Non mi interessava la marca, per cui decisi per un anonimo computer assemblato. Il tizio mi mise davanti un listino da cui potevo scegliere il tipo di CPU, la quantità di memoria, la capienza del disco fisso. Io volevo la potenza. Volevo 4 megabyte di RAM invece di 2. Volevo 33 megahertz. Certo, avrei potuto accontentarmi di 16 megahertz, ma volevo il massimo.

Tu gli dicevi cosa volevi e loro te lo mettevano insieme. Sembra una cosa bizzarra in quest'epoca di internet e di corrieri UPS. Tornavi tre giorni dopo a prenderlo, ma quei tre giorni sembravano durare una settimana. Il 5 gennaio chiesi a mio padre di aiutarmi a portare a casa in macchina quel coso.

Non era solo anonimo, ma anche privo di qualsiasi tocco estetico. Era un semplice blocco grigio. Non avevo comprato *questo* computer perché aveva un aspetto figo. Era una macchina dall'aria estremamente noiosa con uno schermo da quattordici pollici, lo scatolone più economico e ragionevolmente tosto che avevo trovato. Tra parentesi: per «tosto intendo un computer potente che pochi altri avevano. Non vorrei farvelo sembrare un coso poco attraente ma funzionale tipo una station-wagon Volvo. Ma il fatto è che volevo una cosa affidabile e per la quale potessi trovare facilmente gli aggiornamenti di cui avrei avuto senz'altro bisogno.

Il computer montava una versione ridotta del DOS. Io volevo farci girare Minix, la variante di Unix, così lo ordinai e ci volle più di un mese perché il sistema operativo arrivasse fino in Finlandia. Potevi comprarti il libro su Minix in un negozio di informatica, ma - visto che non c'era molta richiesta per il sistema operativo - dovevi ordinare il software in libreria. Il prezzo? 169 dollari più le tasse più il cambio più chissà che altro. Mi sembrava un prezzo mostruoso all'epoca. E sinceramente non ho ancora cambiato idea. Quel mese sprecato mi sembrò lungo sei anni. Ero ancora più frustrato di quanto fossi stato nei mesi in cui stavo aspettando di comprare il PC.

Ed eravamo in pieno inverno. Ogni volta che uscivi dalla tua stanza e ti avventuravi nel mondo esterno, rischiavi di farti travolgere da una vecchietta che avrebbe dovuto starsene a casa a preparare la minestra di cavoli e a guardare l'hockey in TV sferruzzando a maglia invece che ciabattare lungo la Mannerheimintie innevata. Fondamentalmente passai quei mesi a giocare a *Prince of Persia* sul mio nuovo computer. E quando non giocavo leggevo dei libri che mi aiutavano a capire il computer che avevo comprato. Alla fine, un venerdì pomeriggio, mi arrivò Minix: lo installai quella notte stessa. Dovetti infilare a turno sedici dischetti nel computer. Tutto il fine settimana fu dedicato a le la conoscenza del nuovo sistema. Capii cosa mi piaceva (e soprattutto cosa non mi piaceva) in quel sistema operativo. Cercai di compensare i difetti scaricando dei programmi che avevo già usato sul computer dell'università, Alla fine mi ci volle più di un mese per adattare il sistema alle mie esigenze.

Andrew Tanenbaum, il professore di Amsterdam che aveva scritto Minix, voleva che il sistema operativo restasse uno strumento didattico. Quindi lo aveva incasinato apposta, e nemmeno poco. Tra le *patch* disponibili per Minix - ovvero dei

miglioramenti - ce n'era una molto famosa scritta da un hacker australiano di nome Bruce Evans, il Dio del Minix 386. I suoi miglioramenti rendevano Minix molto più utilizzabile su un 386. Prima ancora di comprare il computer avevo seguito i newsgroup dedicati a Minix in rete, per cui sapevo fin dall'inizio che avrei voluto far girare la sua versione migliorata. Ma per questioni di licenza bisognava comprare la versione ufficiale di Minix e poi fare un sacco di lavoro per inserire le patch di Evans. Era una faticaccia.

C'era una serie di funzioni deludenti in Minix. Quella peggiore era l'emulazione di terminale, che era importante perché era il programma che usavo per collegarmi al computer dell'università. Mi affidavo all'emulazione di terminale ogni volta che volevo inserirmi nel computer dell'università per lavorare sulla loro potente unità Unix o anche solo per andare in rete.

Così iniziai a progettare un mio programma di emulazione di terminale. Non volevo progettarlo sotto Minix, ma piuttosto a puro livello hardware. Questo progetto di emulazione di terminale fu anche un'ottima opportunità per imparare come funzionava l'hardware del 386. Come ho già detto, a Helsinki era inverno. E io avevo un computer tosto. La parte più importante del progetto consisteva semplicemente nel capire quali erano le possibilità della macchina e divertirmi un po'.

Dato che programmavo a livello hardware, dovetti iniziare dalla BIOS, il primo codice ROM caricato dal computer. La BIOS legge il floppy o il disco fisso e in questo caso io avevo il mio programma su floppy. La BIOS legge il primo settore del floppy e inizia a eseguirlo. Era il mio primo PC e dovevo imparare come si faceva tutto questo. É una funzione che avviene in «modalità reale». Ma per utilizzare tutta la CPU ed entrare in modalità 32 bit, devi entrare in «modalità protetta». E per riuscirci devi affrontare un sacco di impostazioni complicate.

Quindi per creare un programma di emulazione di terminale in questo modo devi sapere come funziona la CPU. In effetti un motivo per cui scrivevo in linguaggio assembly consisteva nell'imparare qualcosa sulla CPU. Le altre cose che devi sapere sono come scrivere sullo schermo, come leggere gli input da tastiera, come leggere e scrivere sul modem. (A questo punto spero di non essermi perso per strada i miei lettori *non-geek* che si sono coraggiosamente rifiutati di saltare a pagina 69.)

Volevo avere due filoni indipendenti. Uno avrebbe letto al modem e visualizzato a schermo. L'altro avrebbe letto da tastiera e scritto a modem. E ci sarebbero stati due canali in entrambe le direzioni. È una cosa che si chiama *task-switching* (commutazione di processi) e il 386 aveva la capacità hardware per farlo. Mi sembrava una buona idea.

Il mio primo programma di prova era fatto per usare un filone per scrivere la lettera A sullo schermo. L'altro filone scriveva la lettera B. (Lo so che non suona come una cosa sconvolgente.) E lo avevo programmato per ripetere la cosa un tot di volte al secondo. Grazie all'interruzione del timer scrissi il programma in modo che lo schermo si riempisse di AAAAAAAAAA. Poi, all'improvviso, passava alla BBBBBBBBBB. É un esercizio del tutto inutile da qualsiasi punto di vista pratico, ma era un buon modo per vedere se la mia commutazione di processi funzionava. Mi ci volle un mesetto per farlo, perché dovevo imparare tutto strada facendo.

Così alla fine riuscii a cambiare i due filoni, l'AAAAAAAAAA e il

BBBBBBBBBB, in modo che uno leggesse dal modem e scrivesse a schermo e l'altro leggesse da tastiera e scrivesse al modem. Avevo il mio programma di emulazione di terminale.

Quando volevo leggere i newsgroup, inserivo il mio floppy, riavviavo la macchina e leggevo le news dal computer dell'università utilizzando il mio programma. Se volevo fare dei cambiamenti per migliorare il pacchetto di emulazione di terminale, entravo in Minix e lo usavo per programmare.

Ne ero molto fiero.

Mia sorella Sara sapeva del mio grande successo personale. Glielo mostrai e lei guardò le schermate di AAAAAAAA e BBBBBBBB per circa cinque secondi, dopodiché disse: «Bravo e se ne andò non molto impressionata. Capii che era una questione di look. É assolutamente impossibile spiegare a qualcun altro che - se pure una cosa non ha un look sconvolgente - dietro possono succedere grandi cose. È un po'come mostrare a qualcuno un pezzo di strada che hai appena asfaltato. Probabilmente l'unica altra persona che lo vide fu Lars, l'altro specializzando in informatica di lingua svedese che aveva iniziato l'università insieme a me.

Era marzo, forse aprile, e se anche la neve si stava trasformando in fanghiglia sulla Petersgatan, io non lo sapevo (e nemmeno me ne importava granché). Passavo la maggior parte del mio tempo in accappatoio davanti al mio nuovo e poco attraente computer, con pesanti tende nere a proteggermi dalla luce del sole (per non parlare del mondo esterno). Mi davo da fare per pagare le rate del mio PC, che avrebbero dovuto durare per altri tre anni. Non sapevo che quelle rate le avrei pagate solo per un altro anno. Perché in quell'anno avrei scritto Linux, che questa volta non avrebbero visto soltanto Sara e Lars. E a quel punto Peter Anvin, che oggi lavora con me alla Transmeta, avrebbe dato il via a una colletta su internet per pagare il mio computer.

Tutti sapevano che io non facevo un soldo con Linux. E la gente iniziò a dirsi: Ehi, perché non facciamo una colletta per pagare il computer di Linus?

Fu magnifico.

Io non avevo un soldo. Mi è sempre sembrato importante non chiedere soldi, non chiedere la carità, ma il fatto che quel denaro mi sia stato semplicemente regalato... mi fa venire il magone.

Fu così che iniziò Linux. Quando i miei programmi di prova si trasformarono in un pacchetto di emulazione di terminale.

La rivista «Red Herring» mi manda in Finlandia per scrivere un articolo su Oulu, il centro hi-tech emergente che ospita 141 start-up nonostante l'inospitale dislocazione a poche ore d'auto dal Circolo Polare Artico. É una buona opportunità per incontrare i genitori e la sorella di Linus, Sara, a Helsinki.

Il padre, Nils (che tutti chiamano Nicke), mi dà appuntamento nella hall del Sokos Hotel Vaakuna, di fronte alla stazione di Helsinki. È una persona curata, con occhiali spessi e una barbetta alla Lenin. É tornato da poco da Mosca, dove è stato per quattro anni inviato della Finish Broadcasting Company, e ora sta scrivendo un libro sulla Russia e decidendo se accettare o no un incarico a Washington, un posto che non trova particolarmente interessante. Qualche mese fa ha vinto un prestigioso premio giornalistico nazionale, un riconoscimento che - come mi dirà in seguilo la sua ex moglie Anna - «l'ha molto addolcito».

Nel tardo pomeriggio mi porta a fare un giro sulla sua Volvo V40 nel quartiere innevato di Linus, mi indica il massiccio edificio in cui padre e figlio hanno fatto le elementari, passa davanti all'appartamento dei nonni dove Linus ha vissuto nei primi tre mesi di vita e poi mi porta al palazzo con vista sul parco in cui la famiglia risiedette per i successivi sette anni. Nicke passò uno di quegli anni a Mosca a studiare il comunismo, quando Linus aveva cinque anni. Poi mi indica il condominio giallino in cui Linus e sua sorella si trasferirono dopo il divorzio - un videonoleggio porno al pianoterra ha sostituito il negozio d'elettricista di quando Linus era ragazzino - e infine passiamo davanti alla più imponente di queste strutture, il palazzo di cinque piani dove vivevano i nonni materni di Linus e in cui lui nacque. La madre di Linus, Anna, vive ancora lì. Potremmo essere nell'Upper East Side di Manhattan verso la fine di dicembre. Nicke è un tipo divertente, intelligente, autoironico e con una gestualità per molti versi simile a quella del figlio, come quando si appoggia il mento alla mano mentre parla. Hanno anche lo tesso modo di sogghignare. A differenza di suo figlio, è una persona atletica - un atleta socialista - che gioca a pallacanestro, corre cinque miglia al giorno e la mattina presto va a nuotare in un lago ghiacciato. Ha cinquantacinque anni e cammina con la sicurezza atletica di un uomo sulla quarantina. Un'altra differenza rispetto a Linus: Nicke sembra avere una vita sentimentale abbastanza complicata.

Ceniamo in un ristorante affollato nel centro di Helsinki dove Nicke parla di quanto è stato difficile per Linus avere come padre un attivista comunista che teneva spesso dei comizi e per un po'ha anche ricoperto una carica pubblica. Mi spiega che Linus veniva spesso preso in giro per l'estremismo politico del padre che alcuni genitori rifiutavano addirittura di farlo giocare i propri figli. Era per questo, mi dice Nicke, che Linus cercava sempre di prendere le distanze dalla retorica sinistrorsa che aveva caratterizzato la sua infanzia. «Se provavo a parlarne, lui usciva dalla stanza», dice Nicke. «Oppure si impuntava sempre su posizioni opposte. So che Linus a scuola veniva preso in giro perché aveva il padre sbagliato. Il messaggio per me era: Non mettermi in questa situazione difficile.»

Nicke mi porta a casa sua per una birra. Si trova a nord della zona amministrativa centrale, in una serie di edifici costruiti originariamente negli anni Venti per ospitare gli operai. Saliamo le scale fino al suo appartamento e ci togliamo le scarpe nell'ingresso. Il posto ricorda la controcultura dei tardi anni Sessanta, con lampadari in vimini intrecciato, arazzi terzomondisti e piante d'appartamento. Ci sediamo al tavolo della cucina, Nicke versa la birra e iniziamo a parlare dell'essere padre. «Un genitore non dovrebbe pensare di essere lui a fare del figlio ciò che è», dice prendendo il cellulare per chiamare la donna con cui vive. Mi dice che Linus sta iniziando solo ora a leggere i libri di storia che lui ha tentato per anni di propinargli, e che probabilmente non si è mai nemmeno preoccupato di leggere le poesie del nonno.

Chiedo a Nicke se ha mai espresso interesse per la programmazione, se ha mai chiesto a Linus di spiegargli i concetti fondamentali. Mi dice di no. Mi spiega che padri e figli sono persone distinte e che un'intrusione nella passione di Linus sarebbe stata come «un'invasione della sua anima». Mi sembra a suo agio nel ruolo di padre di una persona famosa. In un suo ritratto su un giornale, pubblicato dopo che aveva vinto il premio giornalistico nazionale, dice che anche nei giorni in cui andava a prendere Linus al parco giochi gli altri ragazzini lo indicavano e dicevano: «Guarda, c'è il padre di Linus!»

Sara Torvalds è arrivata, in treno da casa sua, in una cittadina a ovest di Helsinki dove i cartelli stradali sono prima in svedese e poi in finlandese, dove si può permettere un appartamento con una vasca coi piedi di leone e una sauna e dove - con suo grande piacere - per le strade si sente parlare lo svedese e non il finlandese. Mi spiega che lei fa parte di una minoranza all'interno di una minoranza: da ragazza si è convertita al cattolicesimo, un gesto che l'ha relegata, insieme al 10 per cento di cittadini finlandesi non luterani e ha fatto sì che per alcune settimane il padre la diseredasse.

Oggi è venuta a Helsinki per insegnare catechismo a dei ragazzi in un programma sponsorizzalo dal governo. É una persona gradevole e positiva: a ventinove anni trasuda, ancora lo spirito privo di qualsiasi cinismo di una liceale onesta e indaffarata. La pelle chiara e il viso rotondo le conferiscono una vaga somiglianza col fratello maggiore, ma è evidente quanto lei sia più socievole. Pigia continuamente i tasti del suo cellulare per mandare messaggi di testo agli amici che incontrerà più tardi. E con altrettanta frequenza controlla se ha ricevuto delle risposte. Lavora con successo come traduttrice.

È mezzogiorno e Sara mi sta portando a pranzo da sua madre, con delle fermate in varie località della sua infanzia: il parco, la scuola elementare. «I miei genitori erano comunisti tesserati e noi siamo stati cresciuti così, pensando che l'Unione Sovietica fosse una buona cosa. Ci portarono in vacanza a Mosca. La cosa che ricordo di più è un enorme negozio di giocattoli, molto più grande di quelli di Helsinki.» I genitori divorziarono quando Sara aveva sei anni. «Ricordo quando ci dissero che papà se ne sarebbe andato. Io pensai: Va bene. Niente più liti. In effetti era già andato a Mosca per dei lunghi periodi e dunque eravamo abituati a vederlo partire.» Quando compì i dieci anni Sara decise di andare a vivere con suo padre

(che si era trasferito nella vicina città di Espoo), anziché con Linus e la madre. «Non è che non volessi vivere con mamma. Il fatto era che non volevo vivere con Linus. Così avremmo litigato solo nei fine settimana. Litigavamo sempre. Un po'alla volta crescemmo e finimmo per litigare di meno.»

Arriviamo all'appartamento al primo piano della madre e Anna Torvalds è molto felice di vederci. Il suo soprannome è Mikke. Si rifiuta di lasciarmi indulgere all'usanza finlandese di togliersi le scarpe: «Non sia sciocco. Questo sporco è già sporchissimo e lei non potrà certo peggiorare le cose.» É bassa, ha i capelli scuri e la battuta sempre pronta. Pochi secondi dopo il nostro arrivo suona il telefono. Un'agente immobiliare mi vuole mostrare l'appartamento vuoto accanto a quello di Mikke in modo che possa descriverlo a suo figlio e portargli della documentazione una volta tornato negli Stati Uniti, nel caso Linus fosse interessato a comprare un pied à terre *a Helsinki. Entriamo nell'appartamento dove l'agente immobiliare - che assomiglia in lodo inquietante al personaggio interpretato da Annette Bening in American Beauty - ci impone di indossare delle specie di soprascarpe di panno blu prima di fare il giro della casa. Poco dopo, con tono di voce di un 'allegria irritante, dice qualcosa del tipo: «Guardate questa stanza. É perfetta per un arredo d'antiquariato che volete proteggere dalla luce del sole.» Mikke mi lancia uno sguardo da cospiratrice e risponde prendendola in giro: «Ma che modo grazioso di dirci che in questa stanza non entra mai la luce!»*

Torniamo nella cucina di Mikke, la padrona di casa si siede a un tavolo rettangolare coperto da una tovaglia colorata e ci versa il caffè in enormi tazze. Il suo appartamento, come quello dell'ex marito, è un tripudio di libri e oggetti etnici. Ci sono tende Marimekko bianche e nere. In origine in questo appartamento c'erano tre stanze da letto e una cucina. Quando i figli se ne andarono, Mikke si trasferì nella camera grande in cui fino ad allora era stata Sara. Poi smantellò le pareti della stanza di Linus e quelle della sua vecchia stanza da letto per creare una grande zona giorno. Indica uno spazio vuoto e dice: «Era lì che stava il computer. Mi sa che ci dovrò mettere una lapide o qualcosa del genere. Lei cosa ne dice?» Fuma una sigaretta dietro l'altra. È una conversatrice gradevole, con un inglese tanto perfetto che non fa pause nemmeno quando pronuncia frasi molto gergali. Alla parete della sua camera è appesa una grande bandiera sovietica. L'ha regalata a Linus Jouko Vieurmaki, che l'aveva comprata durante una gara internazionale di salto con gli sci. Linus l'aveva tenuta in un cassetto per anni, ma Mikke l'ha appesa sopra il proprio letto.

Mikke tira fuori un album che contiene qualche foto di famiglia. Linus a due o tre anni, in spiaggia, nudo. Linus alla stessa età davanti a un famoso castello nei pressi di Helsinki. Linus da adolescente, magro e grifagno. Mikke al compleanno di suo padre, il professore di statistica. Mi indica sua sorella maggiore e suo fratello. «Lei fa la psichiatra a New York. Lui il fisico nucleare. E io sono la pecora nera. Giusto? Ma sono stata la prima a dare un nipotino a mio padre», dichiara mentre si accende una Gauloise.

Ceniamo in un ristorante che porta il nome di Wilt Chamberlain. Sara consulta il suo telefonino mentre Mikke ordina una serie di caffè. Mikke ricorda come lei e Nicke litigavano per decidere se Linus dovesse o no essere costretto a rinunciare al

proprio ciuccio: si scrivevano dei bigliettini che lasciavano sul ripiano della cucina. Parliamo un po'della scarsa memoria di Linus e della sua incapacità a ricordarsi i volti. «Se vedi un film con lui e il protagonista si cambia la camicia, Linus ti chiede: 'Ma chi è quello?», dice Sara. Poi parliamo di una vacanza di famiglia in tenda e bicicletta in Svezia. La notte passata sul traghetto. La bicicletta di Sara rubata il primo giorno. La tenda montata sull'orlo di un precipizio. Linus che vi si barricava a leggere mentre madre e figlia nuotavano e pescavano. E poi il momento in cui, dopo una terribile tempesta, avevano capito che l'unica cosa che aveva impedito alla tenda di essere spazzata dentro il Mar Baltico era Linus, che vi era rimasto dentro a dormire senza accorgersi nemmeno della furia degli elementi.

Mikke scoppia a ridere mentre ricorda gli anni in cui Linus si nascondeva in camera, avvinghiato al computer. «Nicke mi diceva continuamente: Fallo uscire, fagli trovare un lavoro, ma Linus non mi preoccupava. Non aveva bisogno degli altri. E qualsiasi cosa stesse facendo con il suo computer, erano affari suoi, era la sua cosa, e aveva il diritto di farla. Non avevo idea di cosa si trattasse».

Adesso sulle attività del figlio sa ciò che sanno tutti. Mikke e gli altri membri della famiglia sono costantemente sotto l'assedio dei media, le cui richieste vengono girate a Linus, che di solito risponde dicendo a madre, padre e sorella di rispondere a proprio giudizio. Ma dopo che hanno scritto la propria risposta, loro in genere la mandano a Linus perché la approvi prima di farla avere al giornalista di turno.

Qualche mese fa, quando ho mandato un'e-mail a Mikke per Chiederle cosa ricordasse dell'infanzia di Linus, la sua risposta fu lunga e ben scritta. Aveva intitolato il proprio saggio «Da un piccolo nerd a Linus: storia di un'educazione». Mikke racconta come avesse notato che sin da poppante Linus mostrava lo stesso tipo di determinazione scientifica che caratterizzava suo padre e suo fratello maggiore:

«Quando vedi una persona i cui occhi si illuminano allorché si presenta un problema o questo persiste, una persona che a quel punto non sente più nemmeno la tua voce, non riesce più a rispondere alla più semplice delle domande, si fa totalmente assorbire dall'attività intrapresa, è pronta a dimenticarsi di mangiare e dormire mentre cerca una soluzione e che non si arrende. Mai. Lui - o lei, naturalmente - può essere interrotto (e nel corso della vita quotidiana questo accade spesso), ma appena può riprende il proprio corso di pensieri. Allora lo sai.»

Mikke scriveva della rivalità tra Linus e Sara e delle loro differenze inconciliabili. (Sara: «Non MI PIACE il sapore di funghi/fegato/qualsiasi cosa.» Linus: «SI CHE TI PIACE!») E poi il loro burbero rispetto. «Linus una volta, da piccolo, espresse la propria stima nei confronti della sorella in modo molto succinto. Avrà avuto cinque anni, sette al massimo, quando mi disse con aria molto seria: Vedi, io non penso mai cose nuove. Penso cose già pensate da altre persone. Gli cambio solo di ordine. Sara invece pensa cose che non sono mai state pensate prima.»

Queste reminiscenze potrebbero rivelare che io non penso a tutt'oggi che Linus abbia un talento «speciale», e certamente non «per i computer»: se non fosse quello sarebbe qualcos'altro. In un altro tempo, in un'altra epoca, si sarebbe concentrato su qualche altra sfida, come penso farà in futuro. (Voglio

dire: spero che non si fermerà su Linux per sempre.) Perché secondo me lui non è motivato da «computer», né dalla fama o dalla ricchezza, ma da una sincera curiosità, dal desiderio di superare le difficoltà che gli si parano davanti e di farlo «nel modo giusto» perché lui è così e non si arrenderà mai. Penso di avere già risposto alla domanda su com'era Linus come figlio: facile da crescere, senza dubbio. Aveva bisogno solo di una sfida, e poi lui faceva tutto il resto. Quando iniziò a concentrarsi sui computer, da ragazzino, fu ancora più facile. Come dicevamo sempre io e Sara, basta dare a Linus uno sgabuzzino con dentro un buon computer, passargli un po'di pastasciutta ogni tanto e lui sarà la persona più felice del mondo. A parte la domanda che mi faceva paura mentre lo vedevo crescere: messo com'è, come diavolo farà a incontrare una ragazza carina? Non potevo far altro che ricorrere all'unica vera risorsa di ogni genitore: incrociare le dita. E grazie al cielo ha funzionato! Ha incontrato Love mentre insegnava all'università e quando lei gli ha fatto dimenticare sia il suo gatto sia il suo computer per qualche giorno, è stato immediatamente chiaro che la Natura aveva trionfato.

Spero solo che i Demoni della Fama, non lo distraggano troppo. (La fama sembrerebbe non averlo cambiato, ma è indubbio che si sia addolcito e che ora tenda a parlare alle persone che gli si avvicinano. Fa addirittura fatica a dire di no. Ma sospetto che questo abbia più a che fare con il fatto di essere diventato marito e padre che con la notorietà.

Ed è evidente che sia la madre sia la sorella si tengono a distanza da questa notorietà. È la fine di gennaio 2000, il giorno dopo l'annuncio pubblico del progetto di Linus da parte della Transmeta. All'inizio del pranzo Mikke chiede a Sara: «Oggi il Giornale diceva qualcosa su tu-sai-chi e tu-sai-cosa?»

Quella sera, tornando a casa dal lavoro, Mikke chiede al taxi di aspettare davanti al mio albergo mentre mi consegna una sedia da bambino in pino che devo portare a Patricia. Oltre a una piantina dell'appartamento libero per Linus.

Il mio primo ricordo di Linus che fa qualcosa di notevole.

Credo che fossimo all'inizio del 1992. Ero andato ancora una volta - in bicicletta e senza nessun programma - a trovare Linus nella sua incasinatissima casa. Mentre guardavamo MTV, come al solito, chiesi a Linus come procedeva il suo sistema operativo. Di solito mi rispondeva con qualche frase senza senso. Quella volta mi portò al suo computer (dalla disordinatissima cucina dei Torvalds alla caotica stanza di Linus).

Inserì nel computer la sua username e la password e arrivò al prompt. Mi mostrò alcune funzionalità di base dell'interprete dei comandi. Niente di speciale. Dopo un po'si voltò verso di me sfoggiando il suo tipico ghigno e mi chiese: «Assomiglia proprio al DOS, vero?»

Io annuii. Non ero sbalordito, perché davvero assomigliava moltissimo al DOS. Non c'era nulla di nuovo. Ma avrei dovuto sapere che Linus non sogghignava mai a quel modo senza un buon motivo. Tornò a voltarsi verso il computer, premette qualche tasto funzione e comparve un'altra schermata di login. Un nuovo login e un nuovo prompt. Linus mi mostrò quattro prompt distinti e mi spiegò che in seguito avrebbero potuto essere utilizzati da quattro utenti diversi.

Fu quello il momento in cui seppi che Linus aveva creato qualcosa di magnifico. Ma non è un problema: posso ancora batterlo a biliardo.

Jouko «Avuton» Vierumaki

Per me voleva dire soprattutto che le linee telefoniche erano sempre occupate e che nessuno riusciva a telefonarci. [...] A un certo punto iniziarono ad arrivare delle cartoline da diversi angoli del pianeta. Penso che fu allora che capii che la gente nel mondo reale stava davvero utilizzando ciò che lui aveva creato.

Sara Torvalds

V. La bellezza della programmazione

Non so davvero come spiegare la mia passione per la programmazione, ma ci proverò. Per chi lo fa, è la cosa più interessante al mondo. É un gioco più coinvolgente degli scacchi, un gioco in cui puoi creare le tue regole e il cui risultato finale è qualsiasi cosa tu riesca a fare.

Ma dall'esterno sembra la cosa più noiosa della terra.

Una parte dell'emozione iniziale della programmazione è facile da spiegare: quando dici al computer di fare una cosa, lui la fa. A prova d'errore. Per sempre. Senza lamentarsi.

E questo è già interessante in sé.

Ma la sola cieca obbedienza, per quanto all'inizio sia affascinante, risulta alla lunga una compagnia poco desiderabile. In effetti questa è la parte che ben presto diventa abbastanza noiosa. A rendere tanto coinvolgente la programmazione è il fatto che, per far fare al computer ciò che vuoi, devi capire *come* farlo.

Personalmente sono convinto che l'informatica abbia molto in comune con la fisica. Entrambe si occupano di come funziona il mondo a un livello abbastanza fondamentale. La differenza, naturalmente, è che mentre in fisica devi capire come è fatto il mondo, in informatica sei tu a crearlo. Dentro i confini del computer, sei *tu* il creatore. Controlli - almeno potenzialmente - tutto ciò che vi succede. Se sei abbastanza bravo, puoi essere un dio. Su piccola scala.

Dicendo questo ho appena offeso più o meno il 50 per cento della popolazione mondiale.

Ma è vero. Crei il tuo mondo e le sole cose che limitano ciò che puoi fare sono la potenza della macchina e ormai sempre più spesso - le tue capacità.

Pensate a una casa sull'albero. Potete costruire una casa sull'albero funzionale, con un'apertura a botola, stabile. Ma lutti conoscono la differenza tra una casa sull'albero ben costruita e una bella, che utilizza in modo creativo l'albero su cui poggia. E questione di combinare arte e ingegneria. Questa è una delle ragioni per cui la programmazione può essere tanto coinvolgente e appagante. La funzionalità di un programma spesso è secondaria rispetto al fatto di essere interessante, bello o scioccante.

È un esercizio di creatività.

All'inizio quello che mi ha attratto nella programmazione è stato il processo di scoprire come funzionasse un computer. Uno dei miei piaceri più grandi consistette nell'apprendere che i computer sono come la matematica: ti ci puoi costruire un tuo mondo con delle regole sue. In fisica sei legato alle leggi esistenti. In matematica invece, come nella programmazione, tutto può essere, purché sia coerente. La matematica non è limitata da alcuna logica eterna, ma deve piuttosto essere logica *in sé*. Come potrà dirvi qualsiasi matematico, si possono avere delle equazioni in cui tre più tre sia uguale a due. Di fatto puoi fare tutto ciò che ti pare, ma aumentando il

livello di complessità devi fare attenzione a non creare qualcosa di incoerente all'interno del mondo che hai generato. Perché quel mondo sia bello, non deve contenere alcun errore. La programmazione funziona allo stesso modo.

Uno dei motivi per cui le persone si innamorano dei computer è il fatto che ti consentono di esperire i nuovi mondi che puoi creare e di capire cosa è possibile. In matematica puoi darti alla ginnastica mentale per immaginare ciò che potrebbe essere. Per esempio, quando la maggior parte delle persone pensa alla geometria, pensa alla geometria euclidea. Ma il computer ci ha aiutato a visualizzare geometrie differenti, non euclidee. Con i computer puoi prendere questi mondi artificiali e vederne l'aspetto, Vi ricordate di Mandelbrot? Le immagini frattali basate sulle equazioni di Benoît Mandelbrot? Erano rappresentazioni visive di un mondo puramente matematico che non avrebbe mai potuto essere visualizzato prima dei computer. Mandelbrot aveva creato delle regole arbitrarie per un mondo che non esisteva e che non aveva alcuna rilevanza sulla realtà, ma si finì per scoprire che aveva creato degli schemi affascinanti. Con i computer e la programmazione si possono costruire nuovi mondi e a volte i loro schemi sono davvero belli.

Per la maggior parte del tempo però non fate niente del genere. Scrivete semplicemente un programma per eseguire un dato compito. In quel caso non state creando un nuovo mondo: state risolvendo un problema nel mondo del computer. Il problema viene risolto pensandoci sopra. E solo un certo tipo di persone è in grado di starsene seduta davanti a uno schermo a pensare. I *geek* come me.

Il sistema operativo è la base di tutto ciò che avviene dentro la macchina. E crearne uno è la sfida più grande di tutte. Quando crei un sistema operativo, stai creando il mondo in cui vivono tutti i programmi che girano su un computer: di base stai inventando le regole di ciò che è accettabile, di ciò che si può fare e ciò che non si può fare, Tutti i programmi lo fanno, ma il sistema operativo è quello che sta alla base. È come dettare la costituzione della terra che state creando, mentre tutti gli altri programmi che girano su quel computer sono solo i singoli codici.

A volte le leggi non hanno senso. Ma è proprio il senso ciò che state cercando. Volete essere in grado di guardare la soluzione e capire di essere giunti alla risposta giusta nel modo giusto.

Ricordate quel vostro compagno di scuola che sapeva sempre le risposte giuste? Era più veloce di tutti gli altri, lo era perché non si doveva sforzare. Non si arrovellava su come si *sarebbe dovuto* risolvere quel problema: semplicemente ci pensava nel modo giusto. E una volta sentita la risposta, non si poteva non trovarla perfettamente sensata.

Lo stesso vale per i computer. Potete andare avanti con la forza bruta, in modo stolido, con un approccio del tipo «dai addosso al problema finché non è più tale», oppure potete trovare l'approccio giusto e all'improvviso il problema scompare. Guardate il problema in un altro modo e avete un'epifania: era un problema solo perché lo stavate guardando nel modo sbagliato.

Probabilmente l'esempio più chiaro in proposito non viene dall'informatica ma dalla matematica. Si dice che quando il grande matematico tedesco Carl Friedrich Gauss andava ancora a scuola il suo insegnante si stesse annoiando e così, per tenere occupati gli studenti, assegnò loro un compito: sommare tutti i numeri da 1 a 100. Si

aspettava che i ragazzi ci mettessero tutto il giorno. Ma il giovane matematico dopo cinque minuti se ne uscì con la risposta esatta: 5.050. La soluzione non consiste nel sommare veramente tutti i numeri, perché sarebbe una cosa frustrante e stupida.

Ciò che Gauss aveva scoperto era che sommando 1 a 100 si tiene 101. Poi sommando 2 a 99 si ottiene ancora 101. La somma di3e98dà101. Fino a 50 più 51, che dà ancora a volta 101. In pochi secondi comprese di avere di fronte 50 coppie di 101 e che quindi la risposta era 5.050.

Forse si tratta di una storia apocrifa, ma il punto è chiaro: un grande matematico non risolve un problema prendendo la strada più lunga e noiosa. Vede lo schema reale nascosto dietro una domanda e applica questo schema per trovare la risposta in un modo migliore. Lo stesso vale per Informatica. Certo, potete scrivere un programma che calcola la somma. Sui computer di oggi sarebbe uno scherzo. Ma un grande programmatore saprebbe che la risposta sta semplicemente nell'essere intelligente. Saprebbe scrivere un bel programma per aggredire il problema in un modo nuovo e - alla fine - nel modo giusto.

Non è facile spiegare cosa possa essere tanto affascinante nel picchiare la testa contro il muro per tre giorni perché non sai come risolvere un problema nel modo migliore, nel modo più bello. Ma una volta trovata una strada, è la sensazione più appagante del mondo.

VI.

Alla mia emulazione di terminale crebbero le gambe. La usavo regolarmente per collegarmi al computer dell'università e leggere la posta o partecipare alle discussioni sul newsgroup di Minix. Il problema era che volevo scaricare e caricare delle cose. Il che voleva dire che avevo bisogno di salvarle su disco. E per farlo la mia emulazione di terminale avrebbe avuto bisogno di un driver per il disco. E anche di un driver per il *file system,* per essere in grado di tenere conto dell'organizzazione dei documenti e salvare come file ciò che scaricavo.

Fu a quel punto che rischiai di arrendermi: pensavo che il gioco non valesse la candela. Ma non c'era molto altro da fare. Quella primavera i corsi che frequentavo non erano particolarmente stimolanti. La mia unica attività sociale era l'incontro (o dovrei dire la festa) settimanale dello Spektrum, tutti i mercoledì sera. Da buon animale antisociale, quella divenne la mia unica occasione per fare qualcosa di diverso dal programmare o studiare. Senza questi incontri (o dovrei dire feste) quella primavera sarei stato un eremita, anziché un semieremita. Lo Spektrum era come una cornice prefabbricata per una qualche vita sociale e non credo di essermi perso uno solo di quegli incontri. Erano importanti per me, tanto importanti che a volle l'attesa mi impediva di dormire, nella speranza che avrei potuto dimenticare per una sera la mia mancanza di capacità sociali, il mio naso e l'evidente assenza di una ragazza nella mia vita. Roba da *geek,* insomma.

Quello che sto cercando di dire è che non avevo un cavolo d'altro da fare. E il progetto per il driver disco/file system sarebbe stata una cosa interessante. Così mi

dissi: OK, lo faccio. Scrissi un driver per il disco. E dato che volevo salvare i documenti nel file system di Minix - e dato che in ogni caso il file system di Minix era ben documentato - resi il mio file system compatibile con quello di Minix. Così potevo leggere i file che avevo creato sotto Minix e scriverli sullo stesso disco in modo che Minix fosse in grado di leggere i file che avevo creato con il programma di emulazione di terminale.

Ci volle un sacco di lavoro, una di quelle robe tipo programma-dormi-programma-dormi-programma-mangiasalatini-programma-dormi-programma-docciaveloce-programma. Quando lo finii fu evidente che il progetto si stava trasformando in un sistema operativo. Così smisi di pensarlo come un'emulazione di terminale e iniziai a pensarlo come un sistema operativo. Credo che la transizione avvenne nello stato ipnotico di una di quelle maratone di programmazione. Era giorno o notte? Non me lo ricordo. Il momento prima me ne stavo lì in accappatoio a smanettare su un'emulazione di terminale con delle funzioni extra. Il momento dopo avevo capito che il programma stava accumulando così tante funzioni da trasformarsi in un abbozzo per un nuovo sistema operativo.

Lo chiamavo il mio «gnu-emacs dei programmi di emulazione di terminalo. Gnu-emacs era partito come editor, ma le persone che l'avevano creato lo avevano dotato di moltissime funzioni. Volevano che fosse un editor programmabile, ma poi questo aspetto prese il sopravvento e divenne una sorta di editor infernale. Può fare qualsiasi cosa parte lavarti i piatti, ed è per questo che spesso l'icona di questo editor è un lavandino. È un tour de force di programmazione con molte più funzioni di quelle necessarie a qualsiasi editor. La stessa cosa stava accadendo alla mia emulazione di terminale. Si stava trasformando in qualcos'altro.

```
Da: torvalds@klaava.Helsinki.Fi (Linus Benedict Torvalds)
A: Newsgroup: comp.os.minix
Oggetto: Gcc-1.40 e domanda posix
Message-ID: «1991Jul13,100050.9886@klaava.Helsinki.Fi»
Data: 3 luglio 1991 10:00:50 GMT
Ciao surfisti,
per un progetto su cui sto lavorando (in Minix)sono interessato alla
definizione degli standard posix. Qualcuno mi potrebbe indicare un
formato (preferibilmente) leggibile a macchina delle più recenti regole
posix? Meglio se su un sito ftp.
```

Ecco, questa è la prima prova pubblica che un geek finlandese stava cercando di mettere alla prova i limiti delle proprie capacità informatiche. Gli standard POSIX sono le lunghissime regole per ciascuna delle centinaia di chiamate di sistema in Unix, quello che devi avere per far svolgere al computer le sue operazioni a partire da read, write, open, close. POSIX è un'entità che si occupa degli standard Unix, un'organizzazione composta da rappresentanti delle aziende che vogliono trovare un accordo sulle linee guida comuni. Gli standard sono importanti perché i programmatori possano scrivere applicazioni per un sistema operativo e possano farle girare su più di una versione. Le chiamate di sistema - in particolare quelle importanti - mi davano un elenco delle diverse funzioni necessarie per un sistema operativo. Io poi scrivevo il codice per eseguire ciascuna funzione a suo modo. Scrivendo secondo

gli standard POSIX, il mio codice sarebbe stato utilizzabile da altri.

All'epoca non sapevo che avrei potuto comprare un libro che riportava queste regole direttamente da POSIX, ma la cosa non avrebbe fatto molta differenza. Anche se mi fossi potuto permettere la spesa, ci sarebbe comunque voluto un sacco di tempo per farmi spedire il tutto il Finlandia. Di qui il mio appello per una versione che potessi scaricare liberamente da un sito ftp.

Nessuno mi rispose per dirmi dove trovare gli standard POSIX, così passai al Piano B. Trovai dei manuali per la versione Sun Microsystems di Unix all'università, che utilizzava un server Sun. I manuali contenevano una versione base delle chiamate di sistema che mi poteva bastare per cavarmela abbastanza bene. Potevo leggere sulle pagine del manuale ciò che una chiamata di sistema avrebbe dovuto fare e poi mettermi a implementarla a partire da lì. I manuali non spiegavano come farlo, dicevano solo quali erano i risultati finali. Studiai anche alcune chiamate di sistema del libro di Andrew Tanenbaum e di qualcun altro. Alla fine qualcuno mi mandò gli enormi volumi che contenevano gli standard POSIX.

Ma il mio messaggio e-mail non passò inosservato. Qualsiasi persona con un po'di cognizione di causa (e il sito Minix veniva letto solo da persone con cognizione di causa) avrebbe potuto capire che il mio progetto riguardava un sistema operativo. Perché sennò avrei dovuto aver bisogno delle regole POSIX? Il messaggio solleticò la curiosità di Ari Lemke, un assistente della Helsinki University of Technology (che avrei frequentato se non fossi stato interessato a studiare la teoria). Ari mi mando una simpatica risposta e si offrì di creare una subdirectory nel sito ftp dell'università per quando fossi stato pronto a pubblicarvi il mio sistema operativo, mettendolo a disposizione di chiunque desiderasse scaricarlo.

VII.

Ari Lemke era una persona davvero ottimista. Creò la subdirectory in questione (*ftp.funet.fi*) molto prima che io avessi qualcosa da pubblicarvi. Avevo la password ed era tutto pronto: mi sarebbe bastato collegarmi e caricare le mie cose. Ma mi ci vollero quattro mesi per decidere che avevo qualcosa che valesse la pena di condividere con il mondo, o quantomeno con Ari e gli altri maniaci di sistemi operativi con cui avevo iniziato a scambiarmi delle e-mail.

Il mio obiettivo originario era quello di creare un sistema operativo che potessi usare come sostituto di Minix. Non doveva fare più cose di Minix, ma piuttosto le cose di Minix che mi interessavano più altre che mancavano in quel sistema. Un esempio: in Minix non era solo l'emulazione di terminale a essere pessima; non c'era modo di eseguire una funzione di controllo del lavoro, ovvero eseguire un programma in background mentre non lo usi. E la gestione della memoria era estremamente semplicistica, come peraltro accade ancora oggi nel sistema operativo Macintosh.

Un sistema operativo si crea scoprendo cosa dovrebbero fare le chiamate di sistema e poi scrivendo un programma per implementarle a modo proprio. In linea

generale ci sono circa duecento chiamate di sistema. Alcune possono rappresentare funzioni multiple. Altre sono abbastanza semplici. Alcune delle chiamate di sistema più fondamentali sono davvero complesse e dipendono da parecchie infrastrutture. Prendiamo ad esempio le chiamate di sistema write e read. Bisogna creare un driver di disco per scrivere o leggere qualcosa da disco. Prendiamo open. Bisogna creare tutto il livello del file system che analizza i nomi e identifica la posizione di qualsiasi cosa si trovi sul disco. Mi ci sono voluti dei mesi solo per scrivere la chiamata di sistema open. Ma una volta definita, lo stesso codice poteva essere usato per altre funzioni.

È così che iniziai a sviluppare il sistema. Leggevo gli standard dal manuale SUN OS o dai libri, prendevo le chiamate di sistema a una a una e cercavo di creare qualcosa che funzionasse. Era davvero frustrante.

Il motivo: dato che non succede nulla, in realtà non puoi vedere alcun progresso. Puoi fare dei piccoli programmi di prova che testano quello che hai aggiunto. Ma sono programmi che non fanno nulla di reale. Dopo un po'ti arrendi, non ce la fai più a leggere un elenco di chiamate di sistema. Ormai il sistema è abbastanza completo perché tu voglia eseguire un programma vero. Il primo programma che devi lanciare è una shell, perché senza una shell è abbastanza difficile eseguire qualsiasi altra cosa. E poi la shell contiene già in sé un buon numero delle chiamate di sistema di cui avrai bisogno. Se riesci a eseguirla potrai ottenere un elenco aggiornato delle chiamate di sistema che non hai implementato.

In Unix la shell è come la madre di tutti i programmi. È lì per avviare gli altri binari. (Un binario è un programma di 1 e di 0 che la macchina può leggere. Quando scrivi un programma in un qualsiasi linguaggio informatico, poi compili il codice sorgente e questo diviene un binario.) La *shell* ti permette di entrare nel sistema. Ok, tradizionalmente in un vero sistema Unix il primo programma che segui si chiama *init,* ma *init* in realtà ha bisogno di parecchie infrastrutture per funzionare. È una specie di controllore ili quello che succede. Ma quando non hai niente che funzioni veramente, non ha molto senso lanciare *init.*

Così invece di avviare *init,* la prima cosa che il mio *kernel* fece fu lanciare la *shell.* Avevo implementato più o meno venticinque chiamate di sistema e, come ho già detto, quello era il primo programma che provavo a far girare. La *shell* non l'avevo scritta io. Avevo scaricato su disco un clone della Bourne Shell, una delle *shell* originali di Unix. Era disponibile su internet come *free software* e il suo nome derivava da una pessima battuta. Il tizio che aveva scritto l'originale si chiamava Bourne, così il clone aveva preso il nome di Bourne-Again Shell[4] , comunemente nota come *bash.*

Quando provi a caricare un vero programma da disco, c'è invariabilmente un baco nel driver di disco o nel caricatore, perché la *shell* non capisce cosa sta leggendo. Così stampa un commento aggiornato di ciò che sta facendo. È importante perché è così che puoi scoprire cosa non funziona.

Arrivai al punto in cui il mio programma stava caricando la *shell* e generando una

---

[4] La pessima e intraducibile battuta consiste nel fatto che «Bourne-again» in inglese ha un'assonanza con «born-again», ovvero «rinato» [N.d.t]

stampa di tutte le chiamate di sistema contenute nella *shell* e non ancora implementate. Avviavo la macchina, facevo partire la *shell* e lei mi risputava indietro una cosa del tipo «chiamata di sistema 512 non eseguita». Guardavo giorno e notte le stampe delle chiamate di sistema, cercando di capire quali avevo sbagliato, Ma era molto più divertente che prendere un elenco di chiamate e implementarle. In questo modo vedevi i progressi che stavi facendo.

Era la fine di agosto o l'inizio di settembre quando riuscii a far funzionare la *shell*. Da quel punto in poi fu tutto più facile.

Era questo il bello della faccenda.

Una volta fatta funzionare la *shell,* ero in grado di compilare quasi immediatamente degli altri programmi. La *shell* era più complessa, per esempio, del programma cp (copy) e del programma 1 (per ottenere un elenco delle directory). Tutto ciò che serviva era già compreso dalla *shell,* per cui una volta che questa funzionava passai da 0 a 100 in un microsecondo: tutti i pezzi erano al proprio posto. A un certo punto ebbi addirittura uno di quei momenti del tipo «e-luce-fu», perché fino a quell'istante non c'era stato niente che avesse veramente funzionato.

Sì, provavo un grande senso di soddisfazione. Credo fosse particolarmente importante perché quell'estate non avevo fatto altro che lavorare al computer. E non sto esagerando. Il periodo da aprile ad agosto è il più bello dell'anno in Finlandia. La gente va in barca nell'arcipelago, prende il sole in spiaggia, se ne sta seduta nella sauna del proprio cottage estivo. Io invece non sapevo nemmeno se era giorno o notte, mercoledì o domenica. Quelle pesanti tende nere tenevano fuori il sole e il mondo. In certi giorni (o notti?) caracollavo direttamente dal letto al computer, distanti non più di mezzo metro. Mio padre insisteva con mia madre perché mi trovassi un lavoretto estivo. Ma a lei non importava: non le davo fastidio. Sara era un po'scocciata dal fatto che quando mi collegavo in rete le linee telefoniche erano sempre occupate. Probabilmente lei esporrebbe questo concetto con minore diplomazia. Non esagero quando dico che non avevo in pratica alcun contatto con il mondo al di fuori del mio computer. Ok, magari una volta alla settimana un amico bussava alla mia finestra e se non stavo controllando del codice troppo importante lo invitavo a entrare. (Era sempre un *lui.* Non dimenticate che erano ancora lontani i tempi in cui essere un *geek* sarebbe stato considerato alla moda.) Ci sorbivamo un tè e magari guardavamo un'oretta di MTV nel cucinino. Ora che ci penso, in effetti ricordo di essere uscito qualche volta per bere una birra o fare una partita a biliardo quando qualcuno tipo Juoko (io lo chiamo «Avuton», che significa «lo sterminatore di draghi», ma questa è un'altra storia) veniva a bussare alla mia finestra. Ma sinceramente all'epoca non succedeva nient'altro nella mia vita.

E non mi sentivo affatto un povero sfigato pallido e con la testa tra le nuvole. La *shell* funzionava, il che significava che avevo costruito le fondamenta di un vero sistema operativo. E mi divertivo.

Con la *shell* in funzione iniziai a testare i programmi interni. Poi compilai programmi nuovi a sufficienza per fare veramente qualcosa. Compilavo tutto in Minix, ma spostai la *shell* in una partizione speciale che avevo creato per il nuovo sistema operativo. Lo chiamai Linux.

Sinceramente: non ho mai pensato di renderlo pubblico con il nome di Linux, mi

sembrava una cosa troppo egocentrica. Sapete qual era il nome che avevo scelto per un'eventuale diffusione? Freax (L'avete capita? *Freax* con la x d'ordinanza). In effetti alcuni dei primi file make - i file che descrivono come compilare le sorgenti - riportarono la parola «Freax» per circa sei mesi. Ma non mi interessava granché. A quel punto non avevo bisogno di un nome perché non avevo intenzione di distribuirlo in nessun modo.

VIII.

```
Da: torvalds@klaava.Helsinki.Fi (Linus Benedict Torvalds) A: Newsgroup:
comp.os.minix Oggetto: Cosa vi piacerebbe di più vedere in minix?
Sommario: piccolo sondaggio per il mio nuovo sistema operativo Messaggi-
ID: « 1991Aug25.205708.954l@klaava.-Helsinki.Fi»

Ciao a tutti quelli che usano Minix. Sto realizzando un sistema operativo
free (è solo per hobby, non sarà una cosa grossa e professionale come
gnu) per cloni 386 (486) AT. Ci lavoro da aprile e sta cominciando a
prendere forma. Mi piacerebbe avere la vostra opinione su cosa vi
piace/non vi piace in minix, perché il mio OS gli assomiglia per alcuni
aspetti (lo stesso layout fisico del file system — per motivi pratici — e
altre cose).

Al momento ho fatto girare bash (1.08) e gcc (1.40) e la cosa sembra
funzionare. Il che vuol dire che entro qualche mese avrò qualcosa di
utilizzabile e mi piacerebbe sapere quali sono le caratteristiche che
interessano di più alla gente. Tutti i suggerimenti sono ben accetti, ma
non vi prometto di implementarli :-)

Linus (torvalds@kruuna.helsinki.fi)

PS: Sì, è privo di codice minix e ha un fs multi-thread. NON è portabile
(utilizza il task-switching 386 ecc.) e probabilmente non supporterà mai
dischi fissi non AT, perché io ho solo quello :-(.
```

I più entusiastici maniaci di sistemi operativi nella comunità Minix videro accendersi una scintilla. Non arrivarono molti suggerimenti sulle caratteristiche di Minix, ma ci furono altri tipi di risposte.

```
>Dicci di più! Ha bisogno di una MMU[5]?

Risposta: sì

>Quanto è scritto in C? Che difficoltà ci saranno per la portabilità?
Sulla faccenda della non-portabilità non ti crederà nessuno :-), e io per
primo lo vorrei portare sul mio Amiga.

Risposta: E per la maggior parte in C, ma la maggior parte della gente
non chiamerebbe C quello che io scrivo. Utilizza tutte le caratteristiche
del 386 che sono riuscito a trovare, perché era anche un progetto che mi
serviva per imparare a conoscere il 386. Alcuni dei miei file «C» in
```

[5] MMU: Memory Management Unit, unità della gestione della memoria

```
realtà sono metà assembler e metà C.

Come ho già detto, utilizza una MMU, sia per la paginazione (non ancora
su disco) sia per la segmentazione. É la segmentazione che lo rende
VERAMENTE dipendente dal 386 (ogni processo ha un segmento di 64Mb per
codice & dati — massimo 64 processi in 4Gb. Chiunque abbia bisogno di
processi da più di 64Mb... peggio per lui.)
```

Diverse persone si offrirono addirittura di fare da beta tester.

Alla fine metterlo in rete non fu proprio una decisione. Era così che ero abituato a scambiare i programmi. Quindi l'unica vera decisione consistette nel capire quando avrei osato mostrare il mio sistema ad altre persone. O, per essere più precisi, la domanda era: quando sarà abbastanza buono da non dovermene vergognare?

Quello che volevo in definitiva era avere un compilatore e un vero ambiente in modo da poter creare i programmi da dentro Linux, senza dover usare Minix. Ma mi sentii così fiero quando la *shell gnu* funzionò che mi sentii pronto a mostrare al mondo la mia creatura. E poi volevo delle opinioni.

Quando la *shell* iniziò a funzionare, avevo già compilato qualche rudimentale binario per il sistema operativo. Non ci si poteva fare granché, ma serviva a vedere che il tutto assomigliava a Unix. In effetti funzionava come una versione incasinata di Unix.

Così decisi che l'avrei reso disponibile. Non l'avrei annunciato pubblicamente. Informai una manciata di persone con delle e-mail private (probabilmente saranno stati cinque o dieci al massimo) che l'avrei caricato sul sito ftp. Tra questi c'erano Bruce Evans di Minix e Ari Lemke. Caricai le sorgenti di Linux e qualche binario in modo che ci si potesse fare qualcosa. Dissi alla gente cosa doveva fare per provare a farlo girare. Dovevano avere installato Minix - la versione 386 - e possedere un compilatore GCC. In effetti dovevano avere la mia versione del GCC, così misi a disposizione anche quella sul sito.

C'è un protocollo per numerare le versioni dei programmi. É una questione psicologica. Quando pensi che una versione sia davvero pronta per essere diffusa, la numero 1.0. Ma prima di quel momento, numeri le versioni per indicare quanto lavoro hai bisogno di fare prima di arrivare alla 1.0. Con quest'idea in mente, il sistema operativo che pubblicai sul sito ftp venne numerato versione 0.01. Così tutti avrebbero capito che non ci si poteva ancora fare granché.

E naturalmente ricordo la data: 17 settembre 1991.

Non penso che fossero più di una o due le persone che lo provarono. Dovevano installare il compilatore speciale, creare una partizione pulita per poterla usare come boot, compilare il mio kernel e poi avviare la *shell.* Avviare la *shell* era praticamente tutto ciò che si potesse fare con quella versione. Potevi stamparti le sorgenti, che non erano più di 10.000 righe, meno di 100 pagine se le stampavi in caratteri piccoli. Ora Linux ha qualcosa come 10 milioni di righe.

Una delle ragioni principali per cui pubblicai il sistema operativo fu per dimostrare che non era tutta aria fritta, che avevo davvero fatto qualcosa. Internet è il regno della chiacchiera. A prescindere da quello che fai - che si tratti di sistemi operativi o di sesso - sono troppi quelli che tirano bidoni nel ciberspazio. Così è bello, dopo avere

parlato a un sacco di gente di creare un sistema operativo, poter dire: «Ecco, l'ho fatto. Non vi stavo prendendo per i fondelli. Ecco quello che ho fatto...»

É Ari Lemke, che si occupò di far arrivare il sistema operativo sul sito ftp, detestava il nome Freax. Preferiva l'altro nome di lavorazione che stavo usando - Linux - e per i file creò l'indirizzo pub/OS/Linux. Ammetto di non avere lottato fino alla morte. Ma è stato lui a farlo. Così posso dire più o meno sinceramente di non essere stato egocentrico. Ma pensai, Ok, non è un brutto nome e in futuro potrò sempre dare la colpa a qualcun altro. Esattamente quello che sto facendo adesso.

Come ho già detto, il mio sistema operativo non era molto utile. In primo luogo si bloccava molto facilmente se occupavi tutti la memoria o se facevi qualcosa di strano. Anche se non facevi niente di strano, il sistema operativo si bloccava se lo facevi girare per un po'di tempo. Ma non era pensato per funzionare, in quella fase. Era pensato per essere guardato. E... ehm... ammirato.

Quindi non doveva essere altro che una chicca per le poche persone interessate a creare nuovi sistemi operativi. Dei tecnici. Anzi: un gruppo ristretto all'interno della comunità dei tecnici.

Le loro reazioni furono invariabilmente positive, ma positive in un senso del tipo «sarebbe carino se potesse fare anche questo o «sembra una figata ma sul mio computer non funziona per niente».

Ricordo una e-mail: diceva che il mio sistema operativo gli era piaciuto molto e andava avanti per almeno un paragrafo a dirmi quanto era bello. Poi mi spiegava che gli aveva appena cancellato tutto il disco fisso e che il mio driver del disco doveva avere un problema. Aveva perso tutto il lavoro che aveva fatto, ma era lo stesso molto positivo. Era divertente leggere e-mail di quel genere. Era una relazione sui bachi che lo avevano incasinato.

Era esattamente il tipo di risposte che stavo cercando. Misi a posto alcuni bachi, come quello che causava il blocco quando veniva esaurita la memoria. E feci il grande passo di portare il compilatore GCC sul sistema operativo, in modo che potessi compilare dei piccoli programmi.

Il che voleva dire che gli utenti non avrebbero avuto bisogno di caricare il mio compilatore GCC prima di avviare il sistema operativo.

IX.

Rimpiangete i tempi in cui gli uomini erano veri uomini e si scrivevano da soli i driver per le periferiche?
Annuncio della pubblicazione di Linux versione 0.02

All'inizio di ottobre vi fu il rilascio della versione 0.02, che comprendeva la sistemazione di alcuni bachi e qualche programma aggiuntivo. Il mese dopo rilasciai la versione 0.03.

Mi sarei potuto fermare alla fine del 1991. Avevo fatto un sacco di cose che mi sembravano interessanti. Non funzionava tutto alla perfezione, ma secondo me nel

mondo del software è facile perdere interesse per un progetto una volta che ne hai risolto i problemi fondamentali. Ed era ciò che stava succedendo a me. Il debug di un software non è un'attività divertentissima. Poi accaddero due cose che mi fecero continuare. Primo: distrussi per errore la mia partizione Minix. Secondo: la gente continuava a mandarmi i propri pareri sul sistema.

All'epoca facevo il boot in Linux ma utilizzavo Minix come ambiente principale di sviluppo. Usavo Linux soprattutto per leggere le e-mail e le news dal computer dell'università tramite l'emulazione di terminale che avevo scritto.

Il computer dell'università era sempre occupato, così avevo scritto un programma che lo chiamava automaticamente. Ma in dicembre mi collegai per errore al mio disco fisso invece che al modem. Stavo cercando di collegarmi a /dev/tt1, la linea seriale. Feci un errore e mi collegai in automatico a /dev/hda1, il disco fisso. Il risultato finale fu che sovrascrissi per errore le parti più critiche della partizione in cui avevo Minix. Sì, voleva dire che non potevo più caricarlo.

Fu a quel punto che dovetti prendere una decisione: potevo reinstallare Minix oppure cogliere l'occasione al volo e dire a me stesso che Linux era abbastanza buono da poterlo sostituire. Avrei scritto i programmi per compilare Linux direttamente al suo interno e quando avessi sentito il bisogno di Minix avrei semplicemente aggiunto quella funzione a Linux. Abbandonare l'ambiente di partenza e rendere veramente indipendente un programma è un passo concettuale notevole, tanto che alla fine di novembre rilasciai la nuova versione chiamandola 0.10. Qualche settimana più tardi venne la versione 0.11.

Fu allora che iniziò a esserci davvero un certo numero di persone che lo utilizzavano e ci facevano delle cose. Fino a quel momento avevo ricevuto solo piccole correzioni di bachi, cose da una riga. Ma ora mi mandavano nuove funzioni. Ricordo di avere aggiornato la mia macchina con 8 mega di RAM invece di 4 per risolvere il problema della memoria aggiuntiva. Mi comprai anche un coprocessore in virgola mobile perché avevano iniziato a chiedermi se Linux lo avrebbe supportato. L'hardware in più avrebbe consentito al mio computer di eseguire funzioni matematiche in virgola mobile.

Ricordo che in dicembre ci fu un tizio in Germania che aveva solo 2 megabyte di RAM e stava cercando di compilare il kernel e non riusciva a far girare il GCC perché all'epoca questo compilatore aveva bisogno di più di un megabyte. Mi chiese se Linux poteva essere compilato con un compilatore meno pesante, che non avesse bisogno di tutta quella memoria. Così decisi che anche se io non avevo bisogno di quella funzione in particolare, l'avrei implementata per lui. Si chiama page-to-disk e significa che anche se uno ha solo 2 mega di RAM può simulare di averne di più usando il disco come memoria. Eravamo più o meno a Natale del 1991. Ricordo che il 23 dicembre stavo cercando di far funzionare il page-to-disk. Il 24 dicembre funzionava abbastanza bene, ma ogni tanto mandava in blocco la macchina. Il 25 era a tutto posto. Quella fu la prima funzione che aggiunsi per soddisfare i bisogni di qualcun altro.

E ne ero fiero.

Non dissi nulla alla mia famiglia quando ci riunimmo a casa della mia nonna paterna (Farmor!) per un pranzo a base di prosciutto e aringhe di ogni tipo. Ogni

giorno la comunità degli utenti di Linux diventava più grande e ricevevo e-mail da posti che avevo sognato di visitare, come l'Australia e gli Stati Uniti. Non chiedetemi perché, ma non sentivo il bisogno di parlare di queste cose con i miei genitori, mia sorella e gli altri parenti. Non capivano niente di computer. Probabilmente pensavo che non avrebbero capito cosa stesse succedendo.

Per quanto ne sapevano, tutto ciò che facevo era occupare le linee telefoniche con il mio modem. A Helsinki di notte c'era una tariffa ridotta, per cui io cercavo di fare la maggior parte del lavoro in quelle ore. Ma qualche volta capitava che tenessi occupato il telefono per tutto il giorno. Cercai di ottenere una seconda linea, ma l'edificio in cui si trovava l'appartamento di mia madre era tanto vecchio che non c'erano linee extra e non ne volevano comprare di nuove. Sara all'epoca non faceva altro che parlare al telefono con le sue amiche. Così ogni tanto litigavamo. In modo virtuale. Mentre lei parlava con le amiche io forzavo il modem a comporre il numero in modo che lei sentisse il classico rumore tipo dee-dee-dee-dee-dee. La cosa la disturbava ma almeno in quel modo capiva che avevo assolutamente bisogno di leggere le e-mail. Non ho mai detto di essere stato il migliore fratello del mondo.

Il page-to-disk era una cosa notevole, perché Minix non l'aveva mai fatto. Lo inclusi nella versione 0.12, che rilasciai nella prima settimana di gennaio 1992. La gente cominciò subito a confrontare Linux non solo con Minix, ma anche con Coherent, un piccolo clone di Unix sviluppato dalla Mark Williams Company. Sin dall'inizio, il fatto di aggiungere il page-to-disk portò Linux un passo avanti rispetto alla concorrenza.

Fu a quel punto che Linux decollò. All'improvviso c'erano persone che passavano da Minix a Linux. All'epoca Linux non faceva tutte le cose che faceva Minix, ma faceva la maggior parte delle cose che interessano davvero alla gente. E aveva questa funzione che in molti trovavano davvero utile: con il page-to-disk potevi far girare programmi più grossi rispetto alla tua memoria. Voleva dire che quando esaurivi la memoria potevi prendere un vecchio pezzo di memoria, salvarlo su disco, ricordare dove l'avevi salvalo e riutilizzare quella memoria per il problema che dovevi risolvere. La cosa fece parecchio clamore nelle prime settimane del 1992.

Fu a gennaio che gli utenti di Linux passarono da cinque, dieci, venti - tutte persone a cui potevo mandare delle e-mail, di cui conoscevo i nomi uno per uno - a centinaia di persone non identificabili. Non conoscevo tutti quelli che usavano Linux, e la cosa mi piaceva.

Più o meno in quel periodo c'era una catena di sant'Antonio che girava su internet. Un qualche poveretto di nome Craig stava morendo di cancro e una serie di e-mail ti spiegavano come stargli vicino mandandogli una cartolina. Si scoprì poi che era una specie di scherzo di cattivo gusto. Non credo che Craig sia mai esistito, e tantomeno che abbia mai avuto il cancro. Ma quell'appello mise in viaggio milioni di cartoline. Quindi quando chiesi a chi usava Linux di mandarmi cartoline anziché denaro, stavo in parte scherzando. Era una specie di scherzo del tipo oh-no-un-altro-messaggio-che-chiede-delle-cartoline.

Nel mondo del PC di allora c'era stata una forte tradizione legata allo shareware. Scaricavi un programma e teoricamente avresti dovuto mandare a chi l'aveva creato dieci dollari o giù di lì. Io ricevevo delle e-mail da gente che mi chiedeva se mi

dovevano mandare una trentina di dollari. Dovevo rispondere qualcosa.

Con il senno di poi, immagino che i soldi mi sarebbero tornati utili. Avevo accumulato circa 5.000 dollari in prestiti universitari e dovevo pagare ogni mese 50 dollari di rate per il mio computer. Le mie altre voci di spesa principali erano la pizza e la birra. Ma Linux mi teneva tanto occupato che non uscivo quasi mai, al massimo una volta la settimana. Non avevo bisogno di soldi per portare fuori le ragazze. Avrei potuto usarli per potenziare il computer, ma non era necessario. Probabilmente un figlio migliore di me avrebbe chiesto dei soldi per il suo programma, quantomeno per passare alla mamma - single e lavoratrice - una parte dell'affitto. A me all'epoca non passò nemmeno per la testa. E se volete fatemi pure causa.

Mi interessava di più vedere dove stavano le persone che usavano Linux. Preferivo le cartoline ai soldi. E arrivarono a fiumi: dalla Nuova Zelanda, dal Giappone, dall'Olanda, dagli Stati Uniti. Di solito era Sara a ritirare la posta e la colpì molto il fatto che quel musone di suo fratello all'improvviso ricevesse delle cartoline da amici sparsi per tutto il mondo. Capì per la prima volta che forse in quelle ore in cui tenevo occupato il telefono stavo facendo qualcosa di potenzialmente utile. Le cartoline arrivavano a centinaia e non ho idea di cosa ne sia stato. Devono essere scomparse in uno dei miei traslochi. Avuton dice che io sono «la persona meno nostalgica» che abbia mai conosciuto.

In effetti non volevo ricevere soldi per una serie di ragioni. Quando avevo pubblicato per la prima volta Linux, sentivo di seguire le orme secolari di scienziati e accademici che avevano costruito il proprio lavoro sulle fondamenta di quello altrui - «sulle spalle dei giganti:», come aveva detto Sir Isaac Newton. Condividevo con gli altri il mio lavoro perché fosse loro utile, ma anche perché volevo ricevere i loro commenti (e... ehm... le loro lodi). Non aveva senso far pagare delle persone che avrebbero potuto aiutarmi a migliorare il mio lavoro. Immagino che avrei assunto un approccio ben diverso se non fossi cresciuto in Finlandia, dove chiunque mostri il minimo segno di avidità viene guardato con sospetto, se non con invidia. (Le cose sono un po'cambiate da quando i telefonini Nokia si sono infilati nelle tasche di tutto il mondo, dando una botta di vita ai conti correnti di molti finlandesi.) E certamente avrei affrontato in modo molto differente tutta la questione niente soldi» se non fossi cresciuto sotto l'influenza di un nonno rigorosamente accademico e un padre rigorosamente comunista.

In ogni caso non volevo vendere Linux. E non volevo perderne il controllo, nel senso che non volevo che nessun altro lo vendesse. Lo spiegavo per bene nella nota di copyright che avevo incluso nella prima versione caricata in rete in settembre. Grazie alla Convenzione di Berna del XIX secolo, una persona possiede il copyright di qualsiasi cosa crei, a meno che lo venda. Come proprietario del copyright, ero io a stabilire le regole: potete usare liberamente il sistema operativo, basta che non lo vendiate, e se vi apportate qualsiasi modifica o miglioramento dovete metterli a disposizione di tutti in codice sorgente (e non in binari, che non sono accessibili). Se non siete d'accordo con queste regole, non avete il diritto di copiare il codice o qualsiasi cosa esso contenga.

Prova a pensarci. Impieghi sei mesi della tua vita per un progetto, vuoi metterlo a disposizione di tutti, vuoi farne qualcosa, ma non vuoi che qualcuno se ne approfitti.

Io volevo che la gente lo potesse vedere e che potesse fare dei cambiamenti e dei miglioramenti in base alle proprie preferenze. Ma volevo anche essere certo di trarne un vantaggio: vedere cosa facevano. Volevo poter avere sempre accesso alle sorgenti in modo che se avessero fatto dei miglioramenti li avrei potuti utilizzare anch'io. Pensavo che il modo per portare Linux a sviluppare la migliore tecnologia possibile consistesse nel mantenerlo puro. Se fossero entrati in gioco i soldi, le cose si sarebbero incasinate. Tenendo alla larga i soldi, tieni lontane anche le persone avide.

Anche se a me non interessava chiedere dei soldi per Linux, altre persone non si facevano scrupoli a chiedere delle donazioni quando davano a qualcuno una copia del sistema operativo dopo averla messa su dischetti. A febbraio non era infrequente che qualcuno si presentasse agli incontri di utenti Unix armato di floppy che contenevano Linux. La gente iniziò a chiedermi se potevano far pagare per esempio cinque dollari solo per coprire i costi del dischetto e del proprio tempo. Il problema era che quella era una violazione del mio copyright.

Era giunto il momento di ripensare il mio approccio sulla non vendibilità di Linux. A quel punto in rete si parlava così tanto di Linux che non temevo più che qualcuno potesse prenderselo e scappare con il bottino, cosa che in precedenza mi aveva molto preoccupato. Quantomeno non avrebbero potuto farlo senza suscitare un sacco di reazioni negative. Se qualcuno avesse provato a dirottare Linux e trasformarlo in un progetto commerciale, ci sarebbe stata una forte reazione e una comunità in piena crescita di hacker avrebbe detto qualcosa del tipo: «Ehi, ma quello è Linux! Non lo puoi faro, anche se magari in modo meno educato.

Ormai la macchina si era messa in moto: ogni giorno degli hacker in tutto il mondo condividevano i propri cambiamenti. Stavamo creando collettivamente il miglior sistema operativo in circolazione e niente avrebbe potuto farci deviare dalla nostra traiettoria. Proprio per questo - e per il fatto che Linux era diventato un marchio così riconoscibile - mi sentii più a mio agio con l'idea che qualcuno lo potesse vendere.

Ma prima che pensiate che io sia Mister Beneficenza, lasciate che vi parli di un altro elemento critico che mi portò a questa decisione. Il fatto è che per rendere Linux utilizzabile mi ero basato su una serie di strumenti distribuiti liberamente in internet: anch'io avevo lavorato salendo sulle spalle dei giganti. Il più importante di questi free software era il compilatore GCC, con un copyright in General Public License, la creazione di Richard Stallman universalmente conosciuta come GPL (o «copyleft»). La GPL non parla di soldi. Puoi far pagare anche un milione di dollari se trovi qualcuno disposto a darteli, l'importante è che tu renda pubbliche le sorgenti. E la persona a cui tu dai o vendi le sorgenti ha gli stessi tuoi diritti. É un ottimo tipo di licenza. Ma a differenza di molti estremisti della GPL, che ritengono che ogni nuova innovazione software dovrebbe essere aperta a tutti con una General Public License, io credo che dovrebbe essere il singolo inventore a poter decidere cosa fare della propria invenzione.

Così abbandonai il mio vecchio copyright e adottai la GPL, un documento scritto da Stallman con la supervisione di alcuni avvocati (e dato che c'entrano degli avvocati, è lungo parecchie pagine).

Il nuovo copyright venne incluso nella versione 0.12, ma ricordo di essere rimasto sveglio la notte dopo il rilascio, tormentato dall'idea dell'interesse commerciale che il

sistema avrebbe suscitato. A ripensarci ora tanta preoccupazione sembra ridicola, dato che l'interesse commerciale era relativamente limitato. Qualcosa mi induceva a pensare che dovevo andarci con i piedi di piombo. Una delle mie preoccupazioni era - ed è tuttora - che qualcuno prendesse Linux senza rispettarne il copyright. All'epoca temevo fosse praticamente impossibile fare causa negli Stati Uniti a una persona che ha infranto un copyright. Ancora oggi la cosa mi preoccupa. É facile portare in tribunale qualcuno per questo tipo di reati, ma quello che mi preoccupa è ciò che questa persona può fare finché non viene costretta a smettere.

E poi c'è il timore che delle aziende in posti tipo la Cina possano infrangere il copyright. Non c'è praticamente nulla nel loro sistema legale che impedisca di farlo e in realtà non vale nemmeno la pena di perseguire chi tenta di fare qualcosa di illegale. É quello che hanno cercato di fare le grandi aziende di software e quelle discografiche e non è che abbiano avuto un grande successo. I miei timori sono mitigati dalla realtà dei fatti. Qualcuno potrebbe anche farlo per un po', ma sono le persone che rispettano il copyright, quelle che condividono i propri cambiamenti al kernel, quelle che lo migliorano, che alla fine prevarranno. Fanno parte del processo di aggiornamento del kernel. Al contrario le persone che non rispettano la GPL non potranno avvantaggiarsi degli aggiornamenti e i loro clienti le abbandoneranno. O almeno spero.

In linea generale io vedo i copyright da due punti di vista. Diciamo che c'è una persona che guadagna 50 dollari al mese. Vi aspettereste che paghi 250 dollari di software? Non credo sia immorale per questa persona copiare illegalmente il software e spendere quei soldi per comprarsi da mangiare. Questo tipo di infrazione del copyright è moralmente giusta. Ed è immorale - per non dire di quanto sia stupido - perseguire questo «criminale». Veniamo a Linux: è irrilevante che qualcuno non rispetti la GPL se utilizza il programma per scopi privati. È quando qualcuno cerca di farci sopra dei soldi facili che la cosa diventa immorale, che accada negli Stati Uniti o in Africa. E anche in questo è tutta una questione di sfumature.

L'avidità è sempre una brutta bestia.

X. Minix contro Linux

Tutta quell'attenzione non fu soltanto positiva. Anche se non ho mai amato molto i confronti, fui costretto a difendere Linux quando Andrew Tanenbaum si mise ad attaccare il sistema operativo che stava soppiantando il suo. Siamo entrambi dei *nerd,* per cui avvenne tutto via e-mail.

Chi poteva volergliene se gli stavano girando vorticosamente? Prima che venissero creati i newsgroup dedicati a Linux, io usavo continuamente i newsgroup di Minix per fare degli annunci che riguardavano Linux o per trovare persone a cui interessasse il sistema operativo. Perché mai Andrew avrebbe dovuto gradire la cosa?

Così, tanto per cominciare, gli dava fastidio che io mi infiltrassi nel suo newsgroup. Naturalmente non gli faceva piacere nemmeno che il suo sistema operativo venisse eclissato da questa nuova creatura proveniente dai ghiacci della

Finlandia. E anche il fatto che moltissimi programmatori si stessero unendo a questo progetto non gli faceva fare i salti di gioia. Aveva idee opposte alle mie su come andasse costruito un sistema operativo. All'epoca Andrew faceva parte di una scuola di informatici che sosteneva un approccio microkernel ai sistemi operativi. Aveva realizzato Minix in microkernel e anche Amoeba, il sistema su cui stava lavorando all'epoca, era dello stesso genere.

Si trattava di un movimento molto vivace nei tardi anni Ottanta e nei primi anni Novanta. E il successo di Linux era una minaccia. Così Andrew continuava a mandare al newsgroup messaggi assai spiacevoli.

La teoria alla base del microkernel parte dall'idea che i sistemi operativi sono complessi. Per cui si cerca di eliminare parte della complessità con una forte modularizzazione. Il principio dell'approccio microkernel è che il kernel - che si trova alla base della base della base - dovrebbe fare il meno possibile. La sua funzione principale è comunicare. Tutte le altre cose che un computer offre sono servizi disponibili tramite i canali di comunicazione microkernel. Nell'approccio microkernel lo spazio dei problemi va suddiviso al punto che nessuna sua parte sia complessa.

Io lo trovavo un approccio stupido. É vero, semplifica ogni singola componente. Ma le interazioni lo rendono molto più complesso di quanto potrebbe essere se molti servizi fossero inclusi nel kernel stesso, come avviene in Linux. Pensate al vostro cervello. Ogni componente è semplice, ma le interazioni tra le componenti creano un sistema estremamente complesso. E il problema «l'intero-è-più-gran-de-delle-parti». Se prendete un problema, lo dividete a metà e dite che le due metà sono il 50 per cento meno complesse del tutto, state ignorando il fatto che dovete aggiungere la complessità della comunicazione tra le due metà. La teoria alla base del microkernel voleva che si dividesse il kernel in cinquanta parti indipendenti, e sosteneva che ciascuna parte fosse un cinquantesimo della complessità del tutto. Ma nessuno teneva conto del fatto che la comunicazione tra le componenti è in realtà più complessa del sistema originale, anche senza tenere conto del fatto che le componenti non sono comunque così semplici.

Questa è l'argomentazione principale contro il microkernel. La semplicità che si cerca di ottenere è una falsa semplicità.

Linux era nato molto più piccolo e molto, molto più semplice. Non implementava la modularità, per cui si potevano fare un sacco di cose in modo più diretto rispetto a quanto accadeva in Minix. Uno dei problemi originari che avevo avuto con Minix era che se avevi cinque programmi diversi che giravano contemporaneamente e tutti quanti volevano leggere cinque file diversi, il processo sarebbe stato serializzato. In altre parole avresti avuto cinque processi diversi che inviavano delle richieste al file system: «Posso leggere il file x, per favore?» Il gestore del file system che si occupa di leggere queste richieste ne prende una e la rimanda indietro, poi prende quella successiva e la rimanda indietro e così via.

Sotto Linux, che è un kernel monolitico, hai cinque diversi processi che eseguono ciascuno una chiamata di sistema al kernel. Il kernel deve stare molto attento che non si confondano l'una con l'altra, ma la scalabilità verso un qualsiasi numero di processi che possono fare ciò che vogliono è molto naturale. Il che rende Linux molto

più veloce ed efficiente.

Un altro problema di Minix era che potevi avere le sorgenti ma le licenze non ti permettevano di farci un granché. Prendete uno come Bruce Evans, che aveva fatto un bel po'di interventi chirurgici su Minix e l'aveva reso più utilizzabile. Non poteva incorporare i suoi miglioramenti. Si doveva limitare a fare delle patch. Da un punto di vista pratico era un disastro. Da un punto di vista legale non poteva mettere a disposizione degli utenti un'immagine di boot in modo che questi potessero fare facilmente un aggiornamento. Così gli utenti dovevano seguire un percorso a tappe per ottenere un sistema utilizzabile. Una cosa tremendamente scomoda.

L'unica volta in cui finii per comunicare con Andrew Tanenbaum fu all'inizio del 1992. Immaginate di collegarvi in rete in una mattina gelida e di trovarvi di fronte la versione non tagliata di questo messaggio:

```
Da: ast@cs.vu.nl (Andy Tanenbaum) A: Newsgroup: comp.os.minix Oggetto:
LINUX è obsoleto Data: 29 gennaio 1992 12:12:50 GMT

Sono stato negli USA per un paio di settimane, per cui non ho commentato
particolarmente LINUX (non che avrei detto molto se fossi stato a casa),
ma per quello che vale ho un paio di commenti da fare ora.

Come molti di voi sanno, per me MINIX è un hobby, una cosa che faccio la
sera quando mi stufo di scrivere libri e non ci sono grosse guerre,
rivoluzioni o udienze del senato da seguire in diretta sulla CNN. Il mio
vero lavoro è quello di professore e ricercatore nell'area dei sistemi
operativi.

Dato il lavoro che faccio, credo di sapere qualcosina su dove andranno a
parare i sistemi operativi nei prossimi dieci anni circa. Due aspetti
sono evidenti:
1.MICROKERNEL CONTRO SISTEMA MONOLITICO

La maggior parte dei sistemi operativi più vecchi sono monolitici, il che
vuol dire che l'intero sistema operativo è un unico file a.out che gira
in «modalità kernel». Questo binario contiene la gestione dei processi,
la gestione della memoria, il file system e il resto. Esempi di questi
sistemi sono UNIX, MS-DOS, VMS, MVS, OS/360, MULTICS e molti altri.
L'alternativa è un sistema basato su microkernel, in cui la maggior parte
dell'OS gira come processi separati, perlopiù fuori dal kernel. I
processi comunicano passando dei messaggi. Il lavoro del kernel consiste
nel gestire il passaggio dei messaggi, l'interrupt, i processi di basso
livello e a volte l'I/O. Esempi di questo tipo di architettura sono
RC4000, AMOEBA, CHORUS, MACH e Windows/NT (di prossima uscita).
Potrei lanciarmi in una lunga storia sui meriti relativi delle due
architetture, ma basti dire che tra le persone che progettano sistemi
operativi il dibattito è fondamentalmente chiuso. Il microkernel ha
vinto, MINIX è un sistema basato su microkernel. Il file system e la
gestione della memoria sono processi separati che girano fuori dal
kernel. Anche i drive I/O sono processi separati. LINUX è un sistema
monolitico. Un grande passo indietro verso gli anni Settanta.

2.PORTABILITÀ

MINIX è stato progettato per essere ragionevolmente portabile, ed è stato
portato dalla linea Intel al 680x0 (Atari, Amiga, Macintosh), allo SPARC
```

e al ns32016. LINUX è legato abbastanza saldamente all'80x86. Ed è la strada sbagliata.
Non fraintendetemi, non è che non sia felice dell'esistenza di LINUX. Mi toglierà di torno tutta quella gente che vorrebbe trasformare MINIX in BSD. Ma in tutta onestà consiglierei a chiunque voglia un sistema operativo *libero* e **MODERNO** di cercare un OS basato su microkernel e portabile, magari qualcosa tipo GNU.

Andy Tanenbaum (ast@cs.vu.nl)

Sapevo di dover difendere il mio onore, per cui risposi:

Da: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds) Oggetto: RE: LINUX è obsoleto Data: 29 gennaio 92 23:14:26 GMT Organizzazione: Università di Helsinki

Be', con un oggetto come questo temo proprio che dovrò rispondere. Mi scuso con gli utenti di minix che hanno sentito parlare anche troppo di linux. Mi piacerebbe riuscire a *ignorare l'amo*, ma... è il momento di farsi una bella litigata!

Nel messaggio «12595@star.es.vu.nl» ast@cs.vu.nl (Andy Tanenbaum) scrive:
>Sono stato negli usa per un paio di settimane, per cui non ho commentato particolarmente LINUX (non che avrei detto molto se »fossi stato a casa), ma per quello che vale ho un paio di commenti da fare ora.
>Come molti di voi sanno, per me MINIX è un hobby, una cosa che faccio la sera quando mi stufo di scrivere libri e non ci sono grosse guerre, rivoluzioni o udienze del senato da seguire in diretta sulla CNN. Il mio vero lavoro è quello di professore e ricercatore nell'area dei sistemi operativi.

Questa sarebbe una scusa per i limiti di minix? Mi dispiace, ma hai perso: io ho molte più scuse di te e nonostante questo linux dà la polvere a minix su quasi tutta la linea. Per non parlare del fatto che la maggior parte del codice migliore di minix sembra essere stata scritta da Bruce Evans.

Risposta 1: tu faresti minix per hobby: guarda chi fa soldi con minix e chi invece regala linux. Poi vienimi a parlare di hobby. Rendi minix disponibile liberamente e una delle mie obiezioni principali scomparirà. Linux è stato decisamente un hobby (ma uno di quelli seri, i migliori) per me: non ci faccio soldi e non fa nemmeno parte dei miei studi universitari. L'ho fatto tutto nel mio tempo libero e con la mia macchina.

Risposta 2 : il tuo lavoro è quello di professore e ricercatore. È una gran bella scusa per alcune delle bufale di minix. Posso solo sperare (e dare per scontato) che Amoeba non faccia schifo come minix.

>1. MICROKERNEL CONTRO SISTEMA MONOLITICO

È vero, linux è monolitico, e concordo sul fatto che il microkernel è più carino. Se ti fossi limitato ad argomentare meno, probabilmente sarei stato d'accordo con quasi tutto quello che hai scritto. Da un punto di vista teorico (ed estetico), linux perde. Se il kernel GNU fosse stato pronto la primavera scorsa, non avrei nemmeno iniziato il mio progetto: il fatto è che non era pronto, e non lo è nemmeno adesso. Linux vince decisamente su questo fronte: è disponibile qui e ora.

>MINIX è un sistema basato su microkernel, [parte cancellata, ma non tanto da confondere le idee] LINUX è un sistema monolitico.

Se questo fosse l'unico criterio per valutare la «bontà» di un kernel, avresti ragione. Quello che però non dici è che minix non si gioca molto bene il microkernel e ha dei problemi con il multitasking reale (nel kernel). Se avessi creato un sistema operativo che avesse dei problemi con un file system multithreading, non condannerei gli altri con tanta leggerezza. Anzi: farei del mio meglio perché gli altri si dimenticassero del mio fiasco.

[sì, lo so che ci sono degli hack per il multithreading in minix, ma sono appunto hack e bruce evans mi dice che ci sono parecchi problemi.]
>2. PORTABILITÀ

«La portabilità è per chi non sa scrivere programmi nuovi»

Io, proprio ora

Il fatto è che linux è più portabile di minix. Sento che stai dicendo: «Cosa?» È vero, ma non nel senso che intendi tu: ho reso linux il più possibile conforme agli standard (senza avere davanti gli standard POSIX). Portare delle cose in linux è in generale molto più facile che portarle in minix.

Concordo sul fatto che la portabilità sia una buona cosa, ma solo quando ha effettivamente un senso. Non ha alcun senso cercare di rendere eccessivamente portabile un sistema operativo: aderire a un API portabile è abbastanza. L'idea stessa di un sistema operativo è quella di utilizzare le caratteristiche hardware e nasconderle dietro uno strato di chiamate di alto livello. È esattamente ciò che fa linux: utilizza un sottoinsieme di caratteristiche del 386 più grande di quanto sembrino fare altri kernel. Naturalmente questo rende il kernel poco portabile, ma rende anche molto più semplice l'architettura. É il compromesso che ha reso possibile linux.
Sono d'accordo anche sul fatto che linux porta la non-portabilità all'estremo: ho preso il mio 386 lo scorso gennaio e linux era in parte anche un progetto per imparare qualcosa sulla macchina. Molte cose avrebbero dovuto essere fatte in modo più portabile, se fosse stato un vero progetto. Ma non sto cercando delle scuse: è stata una decisione di design, e lo scorso aprile, quando ho iniziato il progetto, non pensavo che qualcun altro avrebbe voluto usarlo. Sono felice di poter dire che mi sbagliavo, e dato che le mie sorgenti sono liberamente disponibili a tutti, chiunque può cercare di portarlo, anche se non sarà facile.

Linus

P.S.:>Mi scuso se vi sono sembrato troppo aggressivo. minix non è male se non hai altro a di sposizione. Amoeba può essere una scelta possibile se hai 5-10 386 di riserva, ma non è certamente il mio caso. Di solito non mi tuffo pesce in una litigata in rete, ma quando si parla di linux divento suscettibile :)

A questi due seguirono un po'di altri messaggi. Fu una delle mie poche flame war, come vengono chiamate in gergo le battaglie verbali in rete. Ma avete capito il punto del la questione: c'erano voci contrastanti anche all'inizio. O forse il punto è: state attenti quando ve ne andate là fuori in un forum elettronico. I vostri errori di battitura o grammatica vi perseguiteranno per sempre.

Io e Linus lasciamo le famiglie e gli amici al campo e parliamo per una camminata pomeridiana lungo il corso di un torrente. È il fine settimana del 4 di luglio e abbiamo piantato le tende a Grover Hot Springs, sui monti della Eastern Sierra, in un posto che sembra preso pari pari dalle pagine del «National Geographic». «Oggi è una giornata da Kodak», proclama Linus fermandosi per guardare un prato costellato di fiori selvatici e le vette spettacolari che gli fanno da sfondo. Ci piazziamo in un punto lungo il torrente e gli chiedo di descrivere la sua vita di quei giorni, quando il fascino di Linux si stava diffondendo ben oltre la sua famiglia originaria di entusiasti appartenenti al newsgroup (alcuni di loro Linus li aveva addirittura incontrali di persona).

«Doveva essere fantastico», gli dico. «.Per anni te n'eri stato a smanettare da solo nella tua stanza, con pochissimi contatti con tutto ciò che stava fuori dalla tua CPU. All'improvviso c'è gente da ogni angolo del pianeta che ti dice che hai fatto un lavoro eccezionale. Sei al centro di questa comunità in crescita, che guarda a te come a...»

«Non ricordo che fosse questa gran cosa, per me», risponde Linus. «Non credo lo fosse. Era un po'la cosa a cui pensavo sempre, ma soprattutto perché c'erano sempre problemi da risolvere. In quel senso ci pensavo molto, ma emotivamente non era una gran cosa. Lo era da un punto di vista intellettuale.

«Mi piaceva il fatto che ci fosse un sacco di gente che mi motivava su questo progetto. Pensavo di averne visto la fine, di essere arrivato quasi a completarlo. Ma poi quel punto non arrivava mai perché la gente continuava a darmi nuove ragioni per continuare e nuovi grattacapi su cui arrovellarmi. Ed, era questo a rendere la cosa interessante. In caso contrario probabilmente sarei semplicemente passato a un altro progetto, perché è così che lavoravo, ed era divertente. Ma penso che in realtà mi preoccupasse di più il mio naso e roba del genere», racconta Linus.

Qualche settimana dopo siamo allo Stanford Shopping Center dove Linus non riesce a decidere quali scarpe da jogging comprarsi. «Quanti chilometri corre alla settimana, di solito?», chiede il commesso. Linus sorride: negli ultimi dieci anni è tanto se ha corso un chilometro. L'esercizio fisico non rientrava tra le sue priorità. Ma nei momenti di maggiore debolezza Linus ammette che gli piacerebbe perdere qualche chilo di troppo.

«Tove deve averti convinto a farmi rinunciare alla mia pancetta», scherza dandosi una pacca sul ventre prominente.

«Dille che questa settimana non mi è ancora arrivato il suo assegno», rispondo io.

Poco dopo stiamo girando per il campus di Stanford alla ricerca di un parcheggio. Dopo una mezz'ora facciamo un po'di stretching e iniziamo a correre sui sentierini sterrati al di là del lago asciutto del campus, nei boschi, diretti alla nostra meta: la grande antenna parabolica sulla collina. Non ci arriviamo mai. Aumento il passo e con mia grande sorpresa Linus riesce a starmi alle costole per un migliaio di metri. Poi si ferma, sfiatato. Qualche minuto dopo ci allunghiamo sull'erba in riva al lago.

«Quale fu la reazione della tua famiglia a quello che succedeva con Linux?», gli chiedo. «Dovevano essere abbastanza entusiasti.»

«Non credo che se ne siano accorti veramente», risponde lui «Non voglio dire che la cosa non li interessasse. Ma io programmavo da sempre e le cose, per quanto vedevano loro, non erano cambiate.»

«Be', devi avere detto qualcosa ai tuoi. Magari una volta tuo padre ti stava portando in auto da qualche parte e tu gli hai detto qualcosa del tipo: Ehi, non ci crederai ma sai quella cosa che stavo facendo con il computer? Be', ci sono centinaia di persone che la usano... »

«No», risponde. «Non sentivo il bisogno di condividere questa cosa con la mia famiglia e i miei amici. Non ho mai avuto voglia di parlarne con la gente. Ricordo che Lars Wirzenius quando stavo scrivendo Linux decise di comprare XENIX, la versione SCO di Unix, e cercava di trovare delle scuse con me tipo: Non prenderla nel verso sbagliato. Personalmente non credo mi sia mai importato. Alla fine passò a Linux, ma per me non era una cosa importante. Il fatto che la gente usasse il mio sistema era una cosa carina, ed era magnifico ricevere i loro commenti, ma allo stesso tempo non era troppo importante. Non mi interessava fare proselitismo. Ero fiero che ci fosse della gente che usava il mio codice, ma non ricordo di avere mai avuto la sensazione di voler condividere questa cosa con qualcuno. Non pensavo fosse la cosa più importante del mondo. Non pensavo di fare qualcosa di veramente importante solo perché un centinaio di persone usavano il mio software. Era più un divertimento. Ed è così che lo vivo ancora oggi.»

«Così non volevi nemmeno parlarne con i tuoi genitori, la tua famiglia o i tuoi amici. E non c'era nulla che ti entusiasmasse in ciò che accadeva?», gli chiedo senza nascondere la mia incredulità.

Linus aspetta qualche secondo prima di rispondere. «Non ricordo nemmeno se avevo dei sentimenti, all'epoca.»

Linus si compra una nuova auto, una BMW Z3, una decappottabile a due posti che definisce come l'essenza stessa del divertimento. Di un blu metallico, il colore perfetto per un ragazzo. Sceglie quella tinta perché quel tipo di auto non viene variata in color giallo chiaro, il suo preferito. Il giallo della BMW, mi spiega, «sembra pipì». Per anni ha parcheggialo la sua Pontiac il più vicino possibile all'entrata della sede della Transmeta, nel centro direzionale di Santa Clara. Ma la BMW viene parcheggiata davanti alla finestra del suo ufficio, secondo lui perché così può stare all'ombra. Il vero motivo: Linus può vedere la macchina nuova, mentre lavora al computer.

Poco più di un anno fa abbiamo fatto il nostro primo viaggio a Santa Cruz in una decappottabile, una Mustang bianca che avevo affittato per l'occasione. Durante quella gita Linus si era fermato a dare un'occhiata alle auto sportive parcheggiate davanti alla sauna e alla birreria in cui ci eravamo fermati. Ora stiamo salendo la montagna con la sua auto sportiva. Sorride e infila una curva della Route 17.

«Te la meriti», dico.

Prendo una manciata di CD dal cruscotto.

«Pink Floyd?», chiedo. «Gli Who? Janis Joplin»

«È la musica con cui sono cresciuto. Non ho mai comprato dischi da ragazzino, e in casa c'erano questi. Credo li ascoltasse mia madre, anche se ricordo che la sua grande passione era Elvis Costello.»

È venerdì pomeriggio, uno scintillante venerdì pomeriggio da perfezione californiana con delizie per tutti i sensi: cieli di cobalto per gli occhi, sole per la pelle, la fragranza dell'eucalipto di montagna, il sapore dolce dell'aria pura, i suoni dei Pink Floyd che escono da altoparlanti all'ultimo grido. Certo, a chi ci supera dobbiamo sembrare una specie di cliché postadolescenziale, con i nostri occhiali da sole e i nostri vocalizzi rock d'annata, ma non sono molte le auto che superano la nuova BMW Z3 di Linus.

Parcheggiamo tra veicoli di minor rango a lato delle Highway 1, poco a nord di Santa Cruz, e scendiamo a piedi verso una spiaggia semideserta. Stendiamo i nostri asciugamani al sole caldo e aspettiamo qualche minuto prima di mettere in azione il registratore. Gli chiedo ancora una volta di descrivere il Linus di quei primi tempi.

Lui disegna un rettangolo nella sabbia per rappresentare la sua stanza da letto, poi indica la posizione del letto e quella del computer. «Rotolavo fuori dal letto e controllavo immediatamente l'e-mail», dice muovendo il dito a disegnare quelle rotte mattutine. «In certi giorni non uscivo nemmeno dall'appartamento. Non controllavo l'e-mail solo per vedere se qualcuno mi aveva scritto. Era più una questione di vedere se un dato problema era stato risolto. Del tipo: Che nuovi entusiasmanti problemi abbiamo in menù oggi? Oppure: Qualcuno ha risolto quel problema che avevamo?».

Linus mi dice che la sua vita sociale all'epoca era «patetica». Dato che però gli sembra di essere troppo patetico si corregge: «Diciamo che era appena fuori dal patetico».

«Non è che diventassi proprio un recluso totale», dice, «ma anche quando iniziò la storia di Linux restai l'antisociale che ero sempre stato. Avrai notato che non contatto mai la gente per telefono. È sempre stato così. La maggior parte dei miei amici è gente che vive attaccata al telefono, io no. Prova a immaginare la ricaduta sugli appuntamenti, se non telefoni mai a una donna. Quindi all'epoca avevo qualche amico che veniva a bussare alla mia finestra per quattro chiacchiere e una tazza di tè. Non credo che allora qualcuno potesse notare una qualche differenza del tipo: Ehi, sta facendo qualcosa di veramente importante e un giorno cambierà il mondo. Credo che nessuno abbia mai pensato qualcosa del genere.»

Gli unici eventi sociali per Linus in quel periodo erano gli appuntamenti settimanali della Spektrum, dove incontrava altri laureandi in scienze. E la cosa gli causava molta più ansia di qualsiasi altra faccenda tecnologica.

Che cosa mi preoccupava? La vita sociale in genere. Forse preoccuparsi non è la parola giusta. Era più una questione di impatto emotivo. Roba di ragazze. Linux all'epoca non era così importante per me, in un certo senso non lo è nemmeno ora. In un certo senso riesco ancora a ignorarlo.

«In quei primi anni di università l'aspetto sociale era molto importante. Non è che pensassi di avere la gobba e che la gente mi ridesse alle spalle. Era più che volere degli amici e cose del genere. Uno dei motivi per cui mi piaceva tanto la Spektrum è che ti dava modo di socializzare senza dover essere socievole. Era la mia serata

mondana, tutte le altre le passavo davanti al computer. Emotivamente era una cosa molto più importante di Linux. Linux non mi ha mai sconvolto più di tanto. Non ci ho mai perso il sonno.

«La cosa che mi sconvolgeva davvero - e che ancora oggi mi sconvolge, se è per questo - non è la tecnologia in sé ma l'interazione sociale che le sta attorno. Uno dei motivi per cui il messaggio di Andrew Tanenbaum mi scombussolò non fu tanto il fatto che sollevasse delle questioni tecniche. Se si fosse trattato di chiunque altro non sarei stato troppo a pensarci. Il problema era che quel messaggio lo aveva mandato alla mailing list e mi aveva... Ero preoccupato della mia immagine agli occhi di quelle persone, e lui la stava attaccando.

«Linux mi motivava un sacco anche perché ricevevo tutto quel feedback. Voleva dire che Linux contava qualcosa e che io facevo parte di un gruppo. E ne ero il leader. Chiaro che era importante, ancora più importante di dire a mia mamma e a mio papà che cosa stavo facendo. Mi interessavano di più le persone che usavano Linux. Avevo creato una cerchia sociale e queste persone mi rispettavano. Allora non la vedevo proprio in questi termini, e nemmeno ora. Ma era la cosa più importante. È per questo che ebbi una reazione tanto forte nei confronti di Andrew Tanenbaum».

Il sole comincia a scendere sul Pacifico. È ora di lasciare la spiaggia. Linus insiste perché guidi io - vuole farmi provare l'auto nuova - e perché facciamo la strada più lunga e tortuosa, la Route 9, per tornare alla Silicon Valley.

Linus mi dice che la sua «guerra virtuale» con il creatore di Minix finì per continuare privatamente, via e-mail, perché era diventata troppo virulenta per essere combattuta in pubblico. Si placò per qualche mese. Poi Tanenbaum mandò una mail a Linus segnalandogli la pubblicità di una versione commerciale di Linux apparsa sulla rivista «Byte».

«L'ultima e-mail che ricevetti da Andrew mi chiedeva se era proprio questo ciò che volevo fare, permettere che qualcun altro vendesse il mio lavoro. Io mi limitai a rispondergli con un Sì. Da allora non l'ho più sentito».

Un annetto dopo, quando Linus era in Olanda per la sua prima conferenza, andò all'università in cui insegnava Tanenbaum, sperando di fargli autografare la sua, copia di Sistemi operativi: progettazione e implementazione, il libro che aveva cambialo la sua vita. Aspettò fuori dalla porta dello studio di Tanenbaum, ma lui non si fece vedere. Il professore non era in città. Non si sarebbero mai incontrati.

XI.

La stanza d'albergo era gelida e io me ne restai a letto a tremare, la notte prima del mio primo discorso. In Olanda non riscaldano gli ambienti come in Finlandia e quella stanza non aveva nemmeno i doppi vetri, come se avesse dovuto ospitare dei clienti solo d'estate. Ma il freddo non era l'unica cosa a tenermi sveglio la notte del 4 febbraio 1992. Ero incredibilmente nervoso. I discorsi in pubblico non sono mai stati il mio pane. A scuola ci facevano tenere delle presentazioni delle nostre ricerche - tipo sui topi e cose del genere - e per me era sempre stata una cosa impossibile. Me

ne stavo lì immobile, muto. Poi iniziavo a ridacchiare. E, credetemi, non no mai stato uno che ridacchi più di tanto. Non mi sentivo a mio agio nemmeno quando dovevo andare alla lavagna a far vedere alla classe come avevo risolto un problema.

Ed eccomi lì, a Ede, Olanda, a un'ora di treno da Amsterdam, invitato a parlare per il decimo anniversario del Netherlands Unix Users Group. Volevo dimostrare a me stesso che ce la potevo fare. Un anno prima mi avevano chiesto di parlare davanti a un'associazione del genere in Spagna ma avevo rifiutato perché la paura di parlare in pubblico era stata più forte del desiderio di viaggiare. E all'epoca viaggiare mi piaceva davvero un sacco. (Mi piace ancora, ma non rappresenta più la novità che era per un ragazzino che non era quasi mai uscito dalla Finlandia. I soli posti in cui fossi stato erano la Svezia, dove facemmo qualche vacanza in campeggio, e Mosca, dove andammo a trovare mio papà quando avevo più o meno sei anni.)

Mi dispiaceva abbastanza avere mandato a monte quel viaggio in Spagna, così mi dissi che la prossima volta che mi avessero invitato a parlare da qualche parte avrei accettato. Ma mentre mi rigiravo nel letto e ci stavo ripensando, mi chiedevo se avrei mai superato la mia paura di alzarmi in piedi di fronte a un grande uditorio, temevo di fare scena muta o, peggio, di mettermi a ridacchiare davanti alle 400 persone che erano lì per ascoltarmi.

Lo so, ero un disastro.

Mi dissi le solite cose. Che il pubblico è dalla tua parte, che non sarebbero nemmeno lì se tu non gli piacessi e che ero indubbiamente preparato sull'argomento: le ragioni tecniche per cui avevo deciso di scrivere il kernel di Linux, quelle per cui l'avevo reso open source. Ma non riuscivo a convincermi che il discorso avrebbe avuto successo e la mia mente continuava a sferragliare come un vecchio treno merci. Tremavo tra le coperte. Letteralmente. E non era tutta colpa del freddo.

Come andò il discorso? Be', il pubblico fu molto carino con quel poveraccio evidentemente terrorizzato che si trovarono davanti, avvinghiato alle sue slide di PowerPoint (che Dio sia ringraziato per averci dato Microsoft) come a un salvagente ed esitante nel rispondere alle loro domande. In realtà la parte domanda-risposta fu la migliore. Dopo il mio discorso - se così si poteva chiamare - Marshall Kirk McKusic, un pezzo grosso di BSD Unix, venne a dirmi che aveva trovato interessante il mio intervento. Gli fui talmente grato per quel gesto che volevo mettermi in ginocchio e baciargli i piedi. Ci sono poche cose che ammiro incondizionatamente nel mondo dell'informatica e Kirk è una di queste, soprattutto dopo che fu così gentile con me in quell'occasione.

Il mio primo discorso fu come una terapia d'urto. E così quelli che seguirono. Ma iniziarono a darmi un po'più di fiducia in me stesso.

David continua a chiedermi come cambiò la mia posizione all'interno dell'università con l'esplosione di Linux. Ma io non ricordo di avere mai sentito un professore che lo citava, o di avere visto uno studente che mi indicava a qualche suo amico. Niente del genere. La gente intorno a me, all'università, sapeva di Linux, ma in realtà la maggior parte degli hacker coinvolti nella cosa non vivevano in Finlandia.

Nell'autunno del 1992 mi avevano nominato assistente per i corsi in svedese al dipartimento di informatica. (Andò così: avevano bisogno di un assistente che parlasse svedese per i corsi base. C'erano due soli laureandi in informatica che

parlassero svedese: Lars e Linus. Non è che ci fosse una gran scelta.) All'inizio ero terrorizzato anche solo all'idea di andare alla lavagna e lavorare su un problema, ma non mi ci volle molto per imparare a concentrarmi solo sul materiale senza preoccuparmi del resto. Tra l'altro, tre anni più tardi venni promosso ricercatore, il che voleva dire che anziché essere pagato per insegnare venivo pagato per lavorare nel laboratorio di informatica, ovvero per concentrarmi sullo sviluppo di Linux. Fu l'inizio di una tendenza: farmi pagare da qualcuno per fare Linux. È fondamentalmente quello che succede alla Transmeta.

David: «Ma allora quand'è che Linux ha iniziato a essere una cosa grossa?»

Io: «Non lo è nemmeno adesso»

Va bene, mi correggo. Ha iniziato a diventare una cosa *un po'* più grossa quando è stato chiaro che un sacco di persone utilizzavano Linux come qualcosa di più di un sistema operativo giocattolo. Quando iniziarono a non usarlo solo per gingillarsi capii che se qualcosa fosse andato storto la responsabilità sarebbe stata mia. Quantomeno io mi *sentivo* responsabile. (E mi ci sento ancora adesso.) Nel 1992 il sistema operativo passò dallo statuto di giocattolo a qualcosa di integrato nella vita delle persone, nelle loro attività, nei loro affari.

Il passaggio avvenne nella primavera del 1992, circa un anno dopo che avevo iniziato l'emulazione di terminale, quando la prima versione del sistema di finestre X iniziò a girare sotto Linux. Questo significava che il sistema operativo era in grado di supportare un'interfaccia grafica e che gli utenti potevano lavorare contemporaneamente su più finestre, grazie al progetto di finestre X nato al MIT Fu un grande cambiamento. Ricordo che ci avevo scherzato sopra con Lars, più o meno un anno prima che accadesse, dicendogli che un giorno avremmo fatto girare X e saremmo stati in grado di fare qualsiasi cosa. Non pensavo che sarebbe successo tanto in fretta. Fu un hacker di nome Orest Zborowski che riuscì a portare X in Linux.

Il sistema di finestre X funziona tramite il server X, che fa tutta la grafica. Il server parla con il cliente, gli dice cose tipo: «Voglio una finestra e la voglio di queste dimensioni». La comunicazione passa attraverso un livello chiamato *socket* o - più formalmente - Unix Domain Socket. É così che avviene la comunicazione interna in Unix, ma i socket vengono usati anche per comunicare in internet. Così Orest scrisse il primo livello di socket per Linux solo per portarvi X. L'interfaccia socket di Orest era imbastita in qualche modo e non era integrata con il resto del codice. In quella situazione io accettai la patch perché ne avevamo bisogno, anche se era abbastanza grezza.

Mi ci volle un po'per abituarmi all'idea che avevamo un'interfaccia grafica. Non penso di averla usata quotidianamente per almeno un anno. E oggi non potrei farne a meno. Tengo sempre milioni di finestre aperte mentre lavoro.

Il contributo di Orest non ci mise a disposizione solo le finestre; ci aprì anche una grande porta sul futuro. I socket di domain erano utilizzati per la rete locale che consente il funzionamento del sistema di finestre X. Potevamo partile da quegli stessi socket per far fare a Linux il grande salto vero le reti esterne. Potevamo collegare più computer. Senza le reti, Linux era utilizzabile solo da persone che se ne stavano a casa propria e usavano un modem per collegarsi, oppure facevano tutto in locale. Con grande ottimismo iniziammo a sviluppare le reti Linux sopra i socket originali,

anche se non erano stati affatto pensati per i collegamenti di rete.

Ero tanto convinto che potessimo farcela tranquillamente che feci un grande salto nello schema di numerazione delle versioni. Nel marzo 1992 avevo pensato di rilasciare la versione 0.13. Invece, con l'interfaccia grafica funzionante, pensai che fossimo ormai intorno al 95 per cento del raggiungimento del nostro obiettivo (rilasciare un sistema operativo completo e affidabile, dotato di reti). Così la nuova release fu la 0.95.

Ragazzi, avevo decisamente corso troppo. Che ingenuo...

Le reti sono una brutta bestia e alla fine ci vollero esattamente due anni per metterle a posto, per dare loro una forma che si potesse rendere pubblica. Quando aggiungi le reti all'improvviso introduci una serie di nuovi problemi. Problemi di sicurezza, per esempio. Non sai chi c'è là fuori, né cosa vuol fare. Devi stare molto attento che qualcun altro non ti mandi in crash il computer inviandoti dei pacchetti di spazzatura. Non puoi più controllare chi sta cercando di contattare la tua macchina. E poi molte persone hanno impostazioni di sistema diversissime. Sembrava che saremmo andati avanti così per sempre. Alla fine del 1993 avevamo delle funzioni di rete quasi utilizzabili, anche se alcune persone avevano dei seri problemi a farle girare. Non riuscivamo a gestire reti che non avessero confini a 8 bit.

Dato che ero stato così ottimista nella numerazione della versione 0.95, mi ritrovai nei casini. Nei due anni che ci vollero per arrivare alla versione 1.0 dovemmo fare dei salti mortali con i numeri delle release. Non c'erano molti numeri tra 95 e 100, ma noi continuavamo a rilasciare nuove versioni basate su sistemazioni di bachi o nuove funzionalità. Quando arrivammo alla versione 0.99 dovemmo cominciare ad aggiungere dei numeri per indicare i livelli delle patch, e poi ci affidammo all'alfabeto. A un certo punto avevamo la versione 0.99, livello di patch 15. Poi la versione 0.99, livello di patch 15B e così via. Facemmo tutta la trafila fino al livello di patch 15Z. Il livello di patch 16 divenne la versione 1.0, il punto in cui il software diventava utilizzabile. Questa versione uscì nel marzo 1994 con grandi squilli di fanfare all'auditorium della facoltà di Informatica dell'Università di Helsinki.

Il periodo precedente era stato abbastanza caotico, ma niente poteva fermare la popolarità di Linux. Avevamo il nostro newsgroup internet - comp.os.linux - che era nato dalle ceneri della mia «guerra virtuale» con Andrew Tanenbaum. E la gente ci arrivava a orde. A quei tempi la Internet Cabal, la gente che più o meno gestiva internet, teneva delle statistiche mensili ufficiose di quanti lettori attraeva ciascun newsgroup. Non erano statistiche affidabili, ma era la migliore fonte di informazioni disponibile sulla popolarità del nostro sito: in questo caso ci diceva a quante persone interessava Linux.

Tra tutti i newsgroup quello di maggior successo era sempre stato alt.sex. Non è il mio preferito, anche se un paio di volte ci sono andato per scoprire la causa di tanto bailamme. Io in effetti sono il classico *nerd* un po'asessuato, preferisco giocare con il mio processore in virgola mobile che tenermi aggiornato sulle ultime novità dal fronte del sesso: posizioni rivoluzionarie, limonate acrobatiche o qualsiasi altra cosa si dicesse la gente su alt.sex.

Con le statistiche mensili di Cabal potevo facilmente tenere sott'occhio la popolarità di comp.os.linux. E - fidatevi - lo facevo. (Se per qualcuno posso essere

una specie di eroe popolare, non sono mai stato il figlio dei fiori tecnologico senza macchia, senza paura e senza egoismi che quei giornali allucinanti cercano di fermi sembrare.) Nell'autunno del 1992 le stime per il nostro newsgroup erano nel l'ordine delle decine di migliaia di persone. Tutta quella gente seguiva il newsgroup per vedere cosa stava succedendo, ma non tutti usavano Linux. Ogni mese, quando uscivano le statistiche, c'era una tabella riassuntiva con i quaranta newsgroup più popolari. Se il tuo newsgroup non riusciva a entrare nei primi quaranta, potevi trovare il rapporto completo su un newsgroup speciale «di manutenzione». Di solito a me toccava fare così.

Il newsgroup di Linux continuò a scalare la classifica. A un certo punto entrò nei primi quaranta e io ne fui felice. Era una figata. Ricordo di avere scritto un messaggio abbastanza vanaglorioso a comp.os.linux in cui fondamentalmente elencavo i vari newsgroup dedicati ai sistemi operativi, compreso Minix, e dicevo: «Ehi, guardate un po', siamo più popolari di Windows» Ricordate che all'epoca quelli a cui piaceva Windows non erano ancora in internet. Nel 1993 arrivammo tra i primi cinque. Quella notte andai a letto tutto soddisfatto ed emozionato all'idea che Linux fosse popolare quasi quanto il sesso.

Non si può dire che nel mio angolino del mondo le cose andassero altrettanto bene. Non avevo una vita. In quel periodo, come ho già detto, Peter Anvin aveva organizzato una colletta online da cui giunsero donazioni per 3000 dollari, sufficienti a pagare il mio computer, cosa che feci alla fine del 1993. E per Natale passai a un 486 DX266, che avrei usato per molti anni. Ma la mia vita era tutta lì: mangiare, dormire. Magari un salto in università. Scrivere codici. Leggere un sacco di e-mail. Sapevo che alcuni miei amici ci davano dentro più di me con il sesso, ma non era un problema.

Francamente, anche quasi tutti i miei amici erano degli sfigati.

XII.

Il discorso che avevo tenuto a Ede mi aveva quasi convinto che avrei potuto sopravvivere a qualsiasi cosa, anche a una cosa terrificante come alzarmi in piedi di fronte a un gruppo di completi estranei ed essere al centro della loro attenzione. La mia fiducia in me stesso stava lentamente aumentando anche in altri campi. Ero costretto a prendere delle decisioni veloci a proposito delle correzioni e degli aggiornamenti di Linux e dopo ogni decisione mi sentivo più a mio agio nel ruolo di leader di una comunità in crescita. Le decisioni tecniche non erano mai state un problema: il problema era capire come dire in modo diplomatico a una persona che preferivo i cambiamenti suggeriti da qualcun altro. A volte era una cosa semplice, tipo «Le correzioni di Tizio e Caio non sono male. Perché non le adottiamo?».

Non ho mai pensato di accettare soluzioni che non mi sembrassero le migliori da un punto di vista tecnico. Era un modo per non schierarmi quando due o più programmatori proponevano delle patch in concorrenza. E poi, anche se all'epoca non ci pensavo in questi termini, era anche un modo per portare la gente a fidarsi di

me. E la fiducia vale un tesoro. Quando la gente si fida di te, accetta i tuoi consigli.

Naturalmente devi costruire le fondamenta su cui questa fiducia si possa poggiare. Credo che questa cosa iniziò non tanto quando scrissi il kernel di Linux, quanto piuttosto quando lo pubblicai in internet, aprendolo a chiunque volesse aggiungere funzioni e particolari a proprio piacere, mentre io prendevo le decisioni finali sul nucleo centrale del sistema operativo.

Come non avevo mai progettato per Linux una vita al di fuori del mio computer, non avevo nemmeno mai deciso di essere il leader. Accadde in modo naturale. A un certo punto un gruppo ristretto di cinque sviluppatori iniziò a generare la maggior parte dell'attività nelle aree chiave dello sviluppo. Sembrava sensato che fossero loro a fare da filtro e assumersi la responsabilità della gestione di queste aree.

Imparai abbastanza presto che il modo migliore e più efficace per essere un leader consiste nel lasciare che le persone facciano le cose perché le vogliono fare, e non perché tu vuoi che le facciano. I migliori leader sanno quando si sbagliano e sono capaci di chiamarsi fuori dal gioco. E i migliori leader mettono gli altri in grado di prendere delle decisioni al posto loro.

Lasciate che mi spieghi meglio. La maggior parte del successo di Linux può essere attribuita ad alcuni miei difetti: 1) sono pigro e 2) mi piace ricevere il merito del lavoro degli altri. Se così non fosse, il modello di sviluppo di Linux (o come altro lo chiama la gente) sarebbe ancora limitato a delle e-mail quotidiane tra una mezza dozzina di geek invece che essere una ragnatela intricata di centinaia di migliaia di partecipanti che si affidano a mailing list, convention di sviluppatori e sponsorizzazioni aziendali per portare avanti qualcosa come 4000 progetti, tutti contemporaneamente. Al vertice, per decidere le dispute sul kernel del sistema operativo, c'è un leader il cui istinto non è - e non è mai stato - quello di fare il leader.

E le cose funzionano a meraviglia. Mi sono liberato di alcuni aspetti che non mi interessavano particolarmente. Il primo di questi è stato il livello utente, le parti esterne del sistema con cui gli utenti finali si trovano ad avere direttamente a che fare, in contrapposizione con il codice più interno e profondo. All'inizio qualcuno si propone per gestirlo. Poi il processo di manutenzione di tutti i sottosistemi diviene organico. La gente sa chi si è dato da fare e di chi si può fidare, e le cose succedono da sole. Niente votazioni. Niente ordini. Nessun bisogno di ricontare le schede.

Se per esempio due persone gestiscono tipi simili di driver software, a volte accetto il lavoro di entrambi e alla fine vedo quale viene utilizzato di più. Gli utenti tendono ad affidarsi all'uno piuttosto che all'altro. Oppure, se lasci che entrambi lavorino sul proprio progetto, potrebbero finire per prendere direzioni diverse e i loro contributi potrebbero avere utilizzi molto differenti.

La cosa che stupisce un sacco di gente è che il modello open source funziona per davvero.

Io credo serva capire la mentalità degli hacker dell'universo del free software. (A proposito: io di solito cerco di evitare il termine hacker. In una conversazione personale con dei tecnici io probabilmente mi definirei un hacker, ma negli ultimi tempi questo termine ha finito per significare qualcosa di completamente diverso: ragazzini che non hanno di meglio da fare che irrompere elettronicamente nelle basi dati aziendali, quando farebbero meglio a fare del volontariato alla biblioteca del loro

quartiere o a cercare di portarsi a letto qualcuno.)

Gli hacker - i programmatori - che lavorano su Linux e su altri progetti open source si dimenticano di dormire, di andare in palestra, di andare alle partite dei loro figli e - ebbene sì - ogni tanto anche del sesso, perché adorano la programmazione. E adorano far parte di un tentativo collaborativo globale - Linux è il più grande progetto collaborativo del mondo - finalizzato a costruire la migliore e la più bella tecnologia disponibile a chiunque la desideri. È molto semplice. É divertente.

Ok, con tutta questa svergognata autopromozione sto iniziando a parlare come un comunicato stampa. Gli hacker dell'open source non sono i corrispettivi hi-tech di Madre Teresa di Calcutta. I loro nomi vengono associati ai contributi che forniscono, nelle credit list o negli history file allegati a ciascun progetto. Gli sviluppatori più prolifici attraggono l'attenzione delle aziende che analizzano il codice sperando di trovare - e in seguito assumere - i migliori programmatori. Gli hacker sono motivati in larga parte anche dalla stima che possono guadagnare agli occhi dei loro colleghi grazie ai propri contributi. È un fattore motivazionale importante. Tutti vogliono fare bella figura con i propri colleghi, migliorare la propria reputazione, elevare il proprio status sociale. Lo sviluppo *open source* dà ai programmatori questa possibilità.

Non c'è bisogno di dire che passai la maggior parte del 1993 esattamente come avevo passato la maggior parte del 1992, del 1991 eccetera: avvinghiato a un computer. Ma c'erano dei cambiamenti in arrivo.

Seguendo le orme accademiche di mio nonno, ero diventato assistente all'università di Helsinki e mi era stato assegnato il semestre autunnale del corso in svedese intitolato «Introduzione all'Informatica». Fu così che incontrai Tove. Lei ebbe sulla mia vita un impatto ancora più forte del libro di Andrew Tanenbaum, *Sistemi operativi: progettazione e implementazione*. Ma non starò ad annoiarvi con i dettagli.

Tove era uno dei quindici studenti del mio corso. Si era già laureata in educazione prescolare. Voleva studiare anche i computer, ma non riusciva a star dietro al resto della classe. Alla fine ce la fece.

Era un corso per principianti, al punto che - ricordate che eravamo nell'autunno del 1993, prima del boom di internet - un giorno i compiti a casa che assegnai alla classe consistevano nel mandarmi delle e-mail. Oggi sembra assurdo, ma dissi: «Come compito a casa, mandatemi delle e-mail.»

Le e-mail degli altri studenti contenevano messaggi di prova o banali osservazioni sul corso.

Tove mi chiese di uscire con lei.

Ho sposato la prima donna che mi ha abbordato via internet.

Il nostro primo appuntamento non è mai finito. Tove insegnava all'asilo ed era stata per sei volte campionessa finlandese di karate. Veniva da una famiglia regolare, anche se io descriverei come regolare qualsiasi famiglia meno sballata della mia. Aveva molti amici. Mi sembrò la donna giusta dal primo momento in cui ci ritrovammo insieme (e non vi spiegherò come arrivai a questa conclusione). Nel giro di pochi mesi io e il gatto Randi ci trasferimmo nel suo minuscolo appartamento.

Per le prime due settimane non portai nemmeno con me il computer. Se escludiamo il servizio militare, quelle due settimane erano il periodo più lungo che io

avessi mai passato lontano da un computer da quando avevo undici anni e me ne stavo seduto in braccio a mio nonno. E -nel caso la cosa vi interessi - quel record non è ancora stato battuto. In qualche modo me la cavai (e nemmeno qui vi spiegherò come). Mia madre, le poche volte che la vedevo, mormorava qualcosa a proposito di «un trionfo di Madre Natura». Credo che mia sorella e mio padre fossero semplicemente sbalorditi.

Poco dopo Tove uscì a comprare una gatta per fare compagnia a Randi. Poi ci inserimmo in una piacevole routine di serate da soli o con gli amici e risvegli alle cinque del mattino, in modo che Tove potesse andare a lavorare e io potessi arrivare in università prima che ci fosse altra gente a disturbarmi, per leggere le mie e-mail su Linux.

3 Il reginetto del ballo

I.

La nascita della versione 1.0 portò con sé una novità: il bisogno di relazioni pubbliche. A me sarebbe bastato presentare al mondo la nuova versione come avevo fatto con quelle precedenti. Avrei scritto sul newsgroup qualcosa del tipo «È pronta la versione 1.0. Vedete voi cosa potete farci.» (Ok, magari non proprio con queste parole esatte.)

Un sacco di altra gente pensava che fosse una cosa più grossa. Volevano la versione 1.0 per questioni di marketing. C'erano tutte queste giovani società commerciali che avevano iniziato a vendere Linux. Per loro la versione 1.0 era importante per motivi psicologici, più che tecnici. E io ero d'accordo. Il fatto è che non fai una bellissima impressione quando cerchi di vendere la versione 0.96 di un sistema operativo.

Io volevo che uscisse perché per me era una pietra miliare e perché avrebbe voluto dire che potevo smettere di correggere bachi per un po'e tornare allo sviluppo. Le società e la comunità Linux volevano proporla al pubblico in grande stile.

Avevamo bisogno di una strategia per le relazioni pubbliche. Io non mi sarei assunto personalmente quel compito. Non mi interessava mandare in giro dei comunicati stampa o fare delle dichiarazioni. Erano gli altri a pensare che si dovesse procedere a quel modo, per cui sarebbero stati loro a prendere il testimone. Era così che andavano le cose con Linux. E in qualche modo tutto funzionava alla perfezione.

Lars fu una delle forze motrici che trasformarono la prima release ufficiale in un vero e proprio evento. D'accordo con altre persone, pensava che l'università fosse il luogo più adatto a dare l'annuncio. Avevano ragione. La mia camera da letto era troppo piccola. E tenere la conferenza in un luogo commerciale avrebbe costituito un precedente sbagliato. Così Lars si offrì volontario per coordinare l'evento con l'università. La facoltà di informatica dell'università di Helsinki era abbastanza piccola da permettergli di parlare direttamente con il rettore.

L'università di Helsinki fu felicissima di mettere a disposizione l'aula magna della facoltà di informatica per la presentazione di Linux 1.0. Perché no? Non accade spesso che un'università possa fare notizia sui telegiornali.

Io accettai di tenere un discorso. Non ero terrorizzato come per la conferenza di Ede. OK, ora che mi ci fate pensare alcune cose furono in effetti ancora più difficili.

Come avere mio padre tra il pubblico. E il fatto che ci fosse la TV finlandese. Era la prima volta che avevo la possibilità di vedermi in televisione. I miei genitori vennero tutti e due a sentirmi (anche se sono abbastanza sicuro che non si sedettero

vicini). C’era anche Tove. Era la prima volta che mio padre incontrava Tove, quindi per me era in gioco molto più della presentazione di Linux 1.0. Ero impegnato nei preparativi dell’ultimo minuto per il discorso - tipo assicurarmi che le mie slide fossero a posto - per cui non c’ero quando si incontrarono entrando nell’auditorium. Forse li intravidi con la coda dell’occhio.

In quel discorso - come in quasi tutti i discorsi che avrei tenuto negli anni a seguire - non parlai tanto di tecnologia quanto di *open source*. Fu divertente. Quell’evento fece cambiare opinione su Linux a un po’di gente della facoltà. Prima di allora la facoltà di informatica era stata fiera di Linux e lo aveva incoraggiato con moderazione. Ma dopo la presentazione la gente del dipartimento iniziò a prendere Linux più seriamente. In fondo l’avevano visto al telegiornale.

Nel corso degli anni qualcuno ha sostenuto che l’università stesse cercando di prendersi il merito per Linux. Non è vero. La facoltà mi ha sempre sostenuto molto. Mi avevano addirittura dato un incarico che mi consentiva di lavorare su Linux. E questo agli inizi, quando nessuno poteva dire: «Spingiamo questa cosa perché un giorno sarà famosa in tutto il mondo Allo stesso tempo furono felici di svolgere un ruolo importante nella presentazione. Fu un’ottima occasione per le relazioni pubbliche. So che adesso ci sono più studenti di lingua svedese alla facoltà di informatica, che in precedenza era sempre rimasta all’ombra del Politecnico.

L’invidia del successo è considerata una tipica caratteristica culturale finlandese. E man mano che Linux si diffondeva in tutto il mondo un sacco di gente mi chiedeva se all’università non erano invidiosi di me. Si dimostrò vero il contrario: mi sostenevano moltissimo. Iniziarono molto presto a sbarazzarsi dei terminali X e a installare dei PC con Linux.

La presentazione diede a Linux molta visibilità in Finlandia e si iniziò a parlarne un po’dappertutto. Buona parie dei primi titoli uscirono perché alcuni giornalisti si erano imbattuti per caso in Linux e ne erano stati colpiti. Da un punto di vista commerciale la versione 1.0 non è mai stata una vera sfida per i pezzi grossi. Linux stava conquistando il mercato che fino ad allora era stato di Minix e Coherent. Ma al di fuori di questa comunità non c’era una grande attenzione per Linux, il che non era male. L’attenzione era comunque molto più alta di quanto mi sarei aspettato all’inizio.

In ogni caso, alcuni giornalisti - soprattutto di testate economiche - iniziarono a bussare alla mia porta. Letteralmente. Tove non era felicissima di svegliarsi il sabato mattina e trovare un giornalista giapponese carico di regali - di solito orologi, perché aveva probabilmente sentito da qualche parte che erano una mia passione - e determinato ad ottenere un’intervista. Era ancora meno felice quando io li invitavo a entrare. Lo schema si ripetè per anni, finché non dichiarammo la nostra nuova casa Zona Degiornalistizzata. Nei momenti di maggiore sconsideratezza mi dimenticavo addirittura di dire a Tove che avevo invitato un giornalista a casa nostra. E me ne dimenticavo pure io. Poi il giornalista arrivava e Tove doveva intrattenerlo finché io non rientravo. In quel periodo iniziarono a nascere anche i siti internet di fan, come quello francese che consiste soprattutto in un’imbarazzante galleria - peraltro poco aggiornata - di mie fotografie. Un esempio? Quella in cui sono a un incontro della Spektrum a petto nudo, con una birra in mano e l’aria da macho.

Lasciamo perdere.

Non erano solo i giornalisti o gli hacker di Linux a prestarmi attenzione. All'improvviso gente con il portafoglio gonfio voleva parlarmi delle proprie tecnologie. Da parecchio tempo Unix era considerato un sistema operativo con grandi possibilità, soprattutto grazie alla sua potenza e al multitasking. Così le società interessate a Unix iniziarono a tenere d'occhio anche Linux. Una di queste fu la Novell, che aveva avviato un progetto basato su Linux. Avevano sviluppato un desktop Unix chiamato Looking Class. Non era male, ma era a un punto morto: non utilizzava lo standard dell'epoca, ovvero il Common Desktop Environment.

Nell'agosto del 1994 mi dissero che mi avrebbero pagato per andarli a trovare a Orem, in Utah, e discutere del loro desktop. La Novell mi stava dando la grande occasione di andare in America, per cui dissi loro che avrei accettato se mi avessero pagato anche il viaggio in un'altra città degli Stati Uniti. Nonostante fossi un provincialotto finlandese, avevo il sospetto che Orem - e se è per questo anche Salt Lake City - non fosse granché rappresentativa del resto del paese. Loro mi suggerirono Washington D.C., ma a me non interessava. Me la immaginavo come un'altra capitale qualsiasi. Mi proposero New York, ma pensai che sarebbe stato più interessante andare in California.

Al quartiere generale della Novell era difficile capire quando prendessero seriamente il progetto. Si scoprì poi che non lo prendevano *affatto* seriamente: finirono per abbandonare il progetto e le nove persone che vi erano coinvolte fondarono la Caldera. Ma io mi stavo godendo la mia prima visita negli Stati Uniti, dove sapevo che prima o poi mi sarebbe capitato di vivere. A prescindere dall'impegno di Novell su Linux, gli Stati Uniti sembravano il centro di un universo tecnologico in crescita.

Quel viaggio rappresentò un piccolo terremoto. La prima cosa che mi colpì fu quanto tutto fosse nuovo in confronto all'Europa. La Chiesa Mormone aveva festeggiato il 50 anniversario qualche anno prima della mia visita e così avevano rimesso a nuovo il loro tempio principale. Era di un bianco abbagliante. Venendo dall'Europa, dove tutte le chiese sono antiche e ricoperte dalla patina del tempo, mentre guardavo quella chiesa potevo pensare a una sola cosa: Disneyland. Sembrava un castello delle favole, non una chiesa. E a Orem feci l'errore di provare la sauna dell'albergo. Era una di quelle saune in miniatura, tutta in plastica e poco più calda dell'ambiente esterno. Me ne andai pensando che negli Stati Uniti di saune non ne capivano niente. Avevo nostalgia di casa.

E iniziai anche a capire come girava. Un visitatore che arrivi in Finlandia impara velocemente a non attaccare bottone con gli estranei nei bar. Allo stesso modo capii che nello Utah - e, come appresi più tardi, in tutta l'America - non puoi discutere razionalmente di aborto e di armi da fuoco. Hai un 50 per cento di possibilità di trovarti di fronte qualcuno che ha un approccio molto emotivo all'argomento ed è molto facile scatenare una vera e propria rissa, in Europa non si litiga su questi argomenti. Negli Stati Uniti molte persone sono tanto arroccate sulle proprie posizioni perché hanno sentito troppo spesso l'altra campana, probabilmente in Finlandia ci sono più fucili pro capite che molti altri paesi, ma li usano soprattutto per andare a caccia. Non è una questione molto sentita.

Un’altra cosa che imparai molto velocemente nei miei primi giorni sul suolo americano: la *root bee*[6]*r* fa schifo.

Dopo lo Utah volai a San Francisco e me ne innamorai. Passavo così tanto tempo a camminare in giro per la città che mi presi una scottatura tremenda e dovetti restare chiuso in camera per un giorno intero.

Ricordo di avere attraversato a piedi il Golden Gate Bridge e di avere alzato lo sguardo sulle Marin Headlands, eccitato all’idea di poter salire quelle colline dopo aver attraversato il ponte. Ma quando arrivai sul lato delle Marin avevo perso ogni interesse per qualsiasi passeggiata a piedi. Non avrei mai potuto immaginare che sei anni dopo sarei stato seduto sulla cresta battuta dal vento di quelle stesse Headlands, lo sguardo perso sul Pacifico, sulla baia di San Francisco, sul ponte, sulla nebbia e sulla città stessa, a spiegare tutto questo al registratore di David.

Mi ci sarebbe voluto soltanto un anno per tornare negli Stati Uniti. Tornai per tenere un discorso al DECUS (Digital’s User Group) di New Orleans. C’erano solo quaranta persone all’incontro, così non fu un’ordalia tremenda. Ma la cosa più importante fu che vi incontrai Maddog, alias Jon Hall. Era il responsabile del marketing tecnico di Digital Unix, nonché un vecchio utente Unix. Era stato lui a chiamarmi per quel discorso. Maddog, che è noto per la barba lunga fino al petto e l’assurdo senso dell’umorismo (per non parlare di quanto russa), dirige Linux International, un’organizzazione che lavora per supportare Linux e i suoi utenti. Ed è anche il padrino di mia figlia Patricia.

Un’altra eredità di quell’incontro di New Orleans: Maddog convinse la Digital a darmi un Alpha. Fu così che Linux venne portato su qualcosa di diverso del PC. Prima di allora c’era gente che aveva portato Linux su altre architetture. C’era una versione per 68K, le macchine Motorola 68000 usate da Atari e Amiga. Ma in quei casi Linux non funzionava allo stesso tempo su entrambe le piattaforme. Per far funzionare quella versione dovevi togliere i pezzi che non funzionavano sulla nuova piattaforma e scriverne di nuovi. Con l’Alpha Linux venne portato per la prima volta. Fondamentalmente le stesse sorgenti funzionavano su PC e su Alpha. Basta aggiungere un livello di astrazione in modo che lo stesso codice sia compilato in due modi diversi per funzionare su due architetture differenti. È sempre lo stesso codice, ma finisce per lavorare in architetture diverse.

Quando rilasciammo la versione 1.2 nel marzo del 1995, il kernel era cresciuto fino a comprendere 250.000 linee di codice, la nuova rivista «Linux Journal» dichiarava una tiratura di 10.000 copie e Linux girava su processori Intel, Digital e Sun sparc. Era un grande passo.

---

[6] Ragazzi, deve essere senz’altro una questione di gusti culturali. Credo sia iniziato tutto con i Puritani, che non potevano bere la birra perché contiene alcol. Così inventarono una bevanda che non conteneva alcol e, dato che era fatta con le radici, la chiamarono «*root beer*» [letteralmente: «birra di radici», n.d.t] per far credere alla gente che fosse buona. E dopo dieci generazioni a cui hanno fatto credere che fosse buona, gli americani se la sono bevuta. Oggi gli americani adorano la ro*ot beer* perché hanno dieci generazioni di ingegneria genetica alle spalle.

II.

Siamo nel 1995. Ci sono sempre più versioni commerciali in circolazione e le società Linux sono al centro dell'attenzione. L'università mi ha promosso da assistente didattico ad assistente ricercatore, il che vuol dire più soldi e meno tempo passato a insegnare. Sto lentamente - *molto* lentamente - completando la mia tesi di laurea, che riguarda il problema di portare Linux su architetture differenti. Tove mi ha fatto scoprire lo squash e giochiamo una volta la settimana: siamo una bella coppia.

Ma ecco che sorge un problema. Si scopre che uno squalo di Boston ha registrato il marchio per la parola Linux. Non solo: ha mandato una e-mail al «Linux Journal» e ad altre società Linux chiedendo il 5 per cento delle loro entrate come «ringraziamento per poter usare il marchio.

Quando sentii questa cosa provai una specie di *déjà vu*. Il nome del tizio mi suonava familiare. Controllai l'archivio delle mie e-mail e scoprii che un annetto e mezzo prima mi aveva mandato un messaggio non richiesto in cui mi chiedeva innanzitutto se credevo in Dio e poi diceva di avere un'eccezionale proposta d'affari per me. Questo accadeva prima che lo *spamming* diventasse un'ossessione globale, erano gli anni innocenti prima che qualcuno pensasse di inquinare internet con offerte di trucchetti per diventare ricco in meno di una settimana. No, non avevo risposto a quel tizio, ma dato che era un tipo di cosa insolita per l'epoca, avevo salvato il messaggio.

Così avevamo per le mani una piccola crisi. Eravamo degli hacker. Nessuno aveva pensato a dare un'occhiata al registro dei trademark.

Il tizio non era uno scippatore di trademark professionista. Pare l'avesse fatto solo quella volta. I trademark sono divisi in varie categorie e lui aveva registrato il suo nella categoria informatica. Insieme alla richiesta di un trademark bisogna presentare delle prove, così lui aveva dato all'ufficio che se ne occupava un dischetto su cui sosteneva di avere un programma chiamato Linux.

Si scatenò un certo panico. Tutta la comunità Linux sapeva che avremmo contestato il trademark. Il problema era che non eravamo organizzati per gettarci nella battaglia. Non avevamo nemmeno abbastanza soldi per ingaggiare un avvocato. Nessuna delle società era propensa a metterci quella cifra, che era di 15.000 dollari. (Oggi spendono una cifra del genere soltanto per rifornire le macchinette delle bibite per i dipendenti.) Ma all'epoca erano un bel po'di soldi per una sola azienda. Così il «Linux Journal» decise insieme ad alcune aziende di mettere dei soldi in Linux International, che avrebbe combattuto per il trademark. Linux International era stato fondato in Australia da una persona di nome Patrick D'Cruze che si era trasferito negli Stati Uniti nel 1994 per contribuire alla promozione mondiale di Linux. L'anno della battaglia per il trademark fu quello in cui Maddog divenne direttore esecutivo. Tutti si fidavano di lui. Ed è ancora così.

Io ero in Finlandia a cercare di battere Tove a squash o Avuton a biliardo e non mi interessava farmi coinvolgere in questa cosa. Volevo solo che quell'incubo finisse

com’era iniziato. La mia idea all’epoca era semplicemente quella di liberarsi del trademark, di farlo dichiarare non valido a causa di un suo precedente utilizzo. Il problema fu che il nostro avvocato ci convinse che sarebbe stato uno di spreco forze. Per considerarlo veramente di pubblico dominio, spiegò, avrebbe dovuto essere un termine generico. E Linux all’epoca non era esattamente un termine generico, l’ufficio trademark probabilmente non lo considererebbe tale nemmeno oggi. Poteva perdere la battaglia, ci disse l’avvocato. E anche se avessimo invalidato il trademark, sarebbe potuto arrivare qualcun altro in futuro e registrarlo di nuovo.

La soluzione che suggeriva consisteva nel trasferire il trademark a qualcun altro. Io votai per Linux International, ma ci fu una certa opposizione. Linux International era un’organizzazione giovane e non aveva ancora dato grande prova di sé. Ci si preoccupava che Linux International venisse controllata da interessi commerciali. (Cosa che non è mai accaduta, potrei aggiungere.) C’era anche una forte preoccupazione su chi sarebbe potuto succedere a Maddog se lui si fosse ritirato.

Così tutti puntarono gli occhi su di me. L’avvocato sostenne che le argomentazioni legali sarebbero state più facili se il trademark Linux fosse stato trasferito a me, perché io ero stato il primo a utilizzare quella parola. Scegliemmo questa strategia. Raggiungemmo un accordo informale perché ci sembrava la cosa più facile ed economica da farsi.

Come per la maggior parte degli accordi informali, non vi posso raccontare i particolari. Non che io li abbia mai conosciuti. Ero felicissimo di disinteressarmene.

Quando tornai a controllare l’e-mail che mi aveva mandato quel tizio mi accorsi che non aveva niente a che fare con brevetti e cose del genere. Era evidente che voleva solo parlare con me.

Forse aveva provato a contattarmi per scucirmi dei soldi. Oppure, se mi fossi dimostrato un vero credente e un’anima santa, mi avrebbe semplicemente regalato il trademark. Non lo so.

Posso accettare il fatto che non tutti abbiano una morale impeccabile. Ma la cosa che all’epoca mi irritava di più era che tutto il sistema dei trademark costringesse me, che non avevo fatto nulla di male, a dare battaglia a quel tizio.

La conseguenza di quella scaramuccia fu che io mi ritrovai con il trademark di Linux. Il che significa che quando delle aziende come VA Linux avviano una IPO (offerta pubblica di vendita), i loro prospetti devono menzionare il fatto che l’azienda non possiede nemmeno il trademark su metà del proprio nome. (In quel caso particolare l’azienda era coinvolta nel processo legale di ottenere da me l’approvazione per l’uso della parola Linux.) Ma mi sono abituato a questo genere di cose.

L’episodio del trademark fu solo un problema inatteso per Linux. É una distrazione. Ma appena sistemato, ne sorse un altro: un ingegnere del laboratorio di ricerca della Intel di Portland, Oregon, disse che la sua società stava usando Linux per esplorare nuove architetture. Mi chiese se mi sarebbe piaciuto trasferirmi lì per un contratto di sei mesi.

Io e Tove avevamo parlato in termini generali della possibilità di vivere negli Stati Uniti. Lei sapeva quanto mi era piaciuto il mio breve viaggio laggiù, a parte la root beer. Eravamo d’accordo sul fatto che le opportunità - per non parlare del clima -

erano migliori in America. (Tra l'altro, sono assolutamente convinto che il sistema americano per motivare i dipendenti sia molto più realistico e dia risultati migliori rispetto al modello europeo. In Finlandia, se un lavoratore è molto migliore dei colleghi, gli si danno un po'di soldi in più e lo si tiene tranquillo. In America gli danno molti soldi in più. E funziona.) Quel contratto sembrava un ottimo modo per mettere un piede nella porta, e decidemmo che avrei dovuto cogliere quell'opportunità. Ma avevo qualche dubbio. Non mi piaceva l'idea di lasciare la scuola senza essermi laureato. Qualcosa dentro di me - forse il ricordo del mio nonno accademico - non accettava l'idea di abbandonare l'università. Alla fine i miei sentimenti non ebbero alcuna importanza. Il dirigente di quell'ingegnere decise che sarebbe stato difficile per me ottenere il permesso di lavoro di sei mesi dall'Ufficio Immigrazione americano e dal Servizio Naturalizzazione.

Così restammo a Helsinki. Mentre i leggendari bevitori finlandesi brindavano all'arrivo del 1996, io mi avvicinavo un centimetro alla volta alla linea di meta della laurea. Dovevo solo completare un corso secondario per finire gli esami. E poi dovevo scrivere la tesi. La cosa buffa è che quella fu l'unica volta in cui ricevetti dei meriti accademici per il mio lavoro su Linux, che aveva occupato la maggior parte della mia vita universitaria.

Il 1996 portò con sé un segnale d'allarme. Nell'egualitaria Finlandia dopo che hai fatto un lavoro per tre anni ottieni un aumento di anzianità imposto dallo stato. Quando vidi la mia prima busta paga sentii come un'ondata di panico: avevo lavorato tanto a lungo all'università da ricevere un aumento per anzianità. Avrei passato tutta la mia carriera a lavorare lì? Ero destinato a diventare come mio nonno? Ricordate la descrizione che ne ho dato all'inizio di questo libro? Pelato, sovrappeso e senza odore. Iniziai a guardarmi regolarmente allo specchio. L'attaccatura dei capelli si era ritirata di un paio di millimetri. La mia pancia, che un tempo era stata incavata come quella di un fachiro, aveva acquistato qualche chilo in più. Avevo ventisei anni e per la prima volta nella vita mi sentivo vecchio. Ero all'università da sette anni. Sapevo che sarei riuscito a laurearmi in fretta se mi fossi dato da fare.

La mia figlia di dieci anni Kaley pensa che puoi dire di avere raggiunto l'apice della fama quando qualcuno ti regala un pinguino. Siamo seduti intorno a un falò nella notte limpida della Sierra quando Linus spiega che un gruppo di utenti di Linux di Bristol, Inghilterra, gli ha comprato un pinguino. Kaley non riesce a credere che Linus non sia mai andato a trovarlo. Poi lui aggiunge che in realtà non gli avevano proprio regalato un pinguino, ne avevano sponsorizzato uno a suo nome. E la sponsorizzazione deve essere durata un anno.

Qualcuno solleva lo sguardo dal proprio marshmallow arrostito e fa l'errore di chiedere come è successo che il pinguino sia diventato il simbolo planetario di Linux.

«Il pinguino è stato un'idea mia», dice Tove. «Linus stava cercando di trovare un simbolo per Linux perché c'era gente che continuava a chiedergli: Non dovremmo avere un simbolo? Lui pensava alle cose che aveva visto in passato. Le società Linux avevano già i propri simboli. Una di queste aveva come simbolo un triangolo rosa. Io però sapevo che quello era il simbolo internazionale dei gay, così gli dissi che era stato già preso. Lui disse che gli sarebbe piaciuta una cosa carina, una cosa simpatica.

«Mi vennero in mente i pinguini. Linus era stato morso da un pinguino in uno zoo australiano. A lui piace giocare con gli animali. Anche se sono dei serpenti a sonagli. Quei pinguini allo zoo erano alti una trentina di centimetri e lui infilò un braccio nella gabbia, per giocarci un po'. Muoveva le dita davanti a loro, come fossero state dei pesci. Il pinguino gli si avvicinò, lo morse, si accorse che non era un pesce. Nonostante quel pinguino l'avesse morso, a lui la cosa piacque comunque. Dopo quella storia ebbi la sensazione che avesse sviluppato una passione per i pinguini. Voleva vedere pinguini ovunque fosse possibile.

«Così quando iniziò a cercare un simbolo io gli dissi: Perché non scegli un pinguino, visto che li adori? E lui rispose: Ok, ci penserò sopra.»

É a questo punto che Linus, seduto due o tre posti più in là, scuote il capo.

«No, non è stata un'idea sua», dice. «Si sbaglia.»

È iniziato così. Linus e Tove di solito non si contraddicono. Tove è incredibilmente brava a gestire le responsabilità delle ragazze e della casa - e di un marito famoso - allontanando i giornalisti con le sue capacità di karateka. Linus sembra decisamente contento di darle una mano piegando ogni tanto la biancheria o facendo il cappuccino la mattina. Anche durante uno stressante viaggio in auto di dieci ore, con i classici tiramolla causati da una coppia di bambini, Linus e Tove gestiscono qualsiasi situazione in modo molto rilassato. Pensate all'equivalente matrimoniale di un divano-letto scandinavo ben costruito.

Avevamo toccato un punto di frizione.

Secondo la versione di Linus, Tove poteva in effetti avere accennato vagamente ai pinguini in una fase preliminare della cosa, ma fu in una conversazione con due pezzi grossi di Linux che le creature polari vennero considerate seriamente per la prima volta come mascotte ufficiali del sistema operativo.

Tocca a Tove riproporre la propria versione dei fatti. «Lui pensava che infondo non fosse una buona idea, perché era una mia idea. Continuò a pensare a un possibile simbolo. Poi un giorno eravamo a Boston con Maddog ed Henry Hall. Iniziarono a parlare del simbolo. Io dissi loro: Che ne dite di un pinguino? Vi sembra carino? Loro dissero di sì. Credo che fu questo a far pensare a Linus che infondo poteva essere una buona idea. «Henry Hall disse che conosceva un artista che poteva disegnarglielo, ma la cosa non andò in porto. Poi so che Linus chiese su internet se c'era qualcuno che voleva mandare delle immagini di pinguini.» Scelse la versione di Larry Ewing, un grafico che lavora all'Institute for Scientific Computing della Texas A&M University.

Il disegno non doveva rappresentare semplicemente un pinguino. Linus ne voleva soprattutto uno che avesse un'aria felice, come se si fosse appena scolato una pinta di birra e avesse fatto la migliore scopata della sua vita. E poi doveva avere qualcosa di diverso dagli altri. E così, mentre tutti gli altri pinguini hanno le pinne e i piedi neri, quelli della mascotte di Linux sono arancione. É un po'come se fosse un incrocio tra un pinguino e un'anatra. Come se Duffy Duck fosse andato un po'fuori di testa durante una crociera in Antartide e avesse passato una notte a folleggiare con una squincia del posto.

III.

La notizia della mia decisione di lavorare per la Transmeta Corporation venne accolta dalla comunità Linux con le stesse reazioni mia e di Tove quando comprendemmo come si facevano i bambini e che ne aspettavamo uno per la fine del 1996.

Quando a primavera si sparse la voce che Tove era incinta, i partecipanti più ciarlieri al newsgroup Linux volevano sapere come pensavo di bilanciare le necessità della manutenzione di Linux con quelle di una famiglia. Qualche mese più tardi, allorché si seppe che (finalmente) avrei lasciato l'Università di Helsinki per lavorare alla Transmeta, nella Silicon Valley, il grande dibattito si incentrò su questo problema: avrei potuto tenere fede alla filosofia *open source* in un pericolosissimo ambiente commerciale come l'avevo tenuta in un'istituzione accademica neutrale? La Transmeta era in parte sovvenzionata dal cofondatore di Microsoft Paul Allen, protestò qualcuno; altri sostenevano che doveva essere un trucco escogitato per assumere il controllo di Linux.

Non dico che queste non fossero preoccupazioni comprensibili per dei membri leali della comunità Linux, ma... lasciatemi respirare! Il fatto è che né la nascita di Patricia nel dicembre del 1996 (o quelle di Daniela sedici mesi più lardi e Celeste quarantotto mesi dopo) né il mio lavoro alla Transmeta, che iniziò nel febbraio del 1997, hanno causalo un declino di Linux. Ho sempre pensato che se qualcosa avesse influenzato negativamente il mio lavoro con Linux, avrei fatto i passi necessari per trasmettere tutta la responsabilità a qualcun altro.

Ma sto correndo troppo.

Nella primavera del 1996, proprio quando il tempo stava iniziando a migliorare, diedi l'ultimo esame del mio corso di laurea. Fu più o meno in quel periodo che mi chiamò Peter Anvin, il membro della comunità Linux che tre anni prima aveva organizzato la colletta online per aiutarmi a pagare il mio primo PC. Come tutti quelli che frequentavano il newsgroup di Linux, sapeva che mi stavo per laureare. Lavorava alla Transmeta da circa un anno e disse al suo capo che conosceva questo tizio in Finlandia che avrebbe potuto fare un buon lavoro per la loro azienda. Venne a farmi una breve visita mentre mi trovavo in Svezia a visitare mia madre. Mi parlò di Transmeta, cosa abbastanza difficile perché era tutto coperto da segreto e non mi poteva dire granché. Le voci che giravano tra programmatori dicevano solo che stavano sviluppando dei «chip programmabili». In ogni caso ero molto contento di incontrare Peter di persona.

Una settimana dopo il suo ritorno in California, mi mandò una e-mail in cui mi chiedeva quando sarei andato a trovarli. Era una storia molto diversa dall'esperienza che avevo avuto con la Intel un anno prima, quando un ingegnere mi voleva mettere sotto contratto ma la cosa non era andata in porto per questioni burocratiche.

In ogni caso sarebbe stato divertente farsi un viaggetto in California, pensai.

Era il mio primo colloquio di lavoro. Non avevo un curriculum vitae. Non sapevo

cosa stessero facendo alla Transmeta. Dovevo andare all’estero.

Ero più preoccupato dalle implicazioni di un trasferimento negli Stati Uniti che dal fatto di ottenere il lavoro così non pensai mai a quegli incontri come a dei colloqui di lavoro. L’importante mi sembrava fosse capire cosa stesse facendo quella gente. Era una situazione abbastanza strana.

Dopo il primo giorno tornai nel mio albergo di fronte al quartiere generale della Transmeta. Ero confuso dal jet-lag, mi sembrava tutto interessante, ma pensavo anche che quelli della Transmeta fossero dei pazzi. In quel momento nell’azienda non c’era un grammo di silicio. Niente hardware. Tutto veniva fatto con un simulatore e la dimostrazione della simulazione che si caricava sotto Windows 3.11 e faceva partire il solitario fece ben poco per convincermi che sarebbe mai successo qualcosa. Dopo il primo giorno mi chiedevo se non fosse tutta una perdita di tempo. Ricordo benissimo di avere pensato: probabilmente non ne uscirà niente, né un’innovazione tecnologica per la Transmeta né un lavoro per me.

Ci dormii sopra. In realtà non dormii un granché. Me ne restai a letto a pensare ai piani della Transmeta. Poi iniziai a sognare a occhi aperti di avere un giardinetto con una palma. Poi mi misi a ruminare su ciò che avevo visto nel simulatore. Fu una notte memorabile, ma niente in confronto al terrore glaciale di Ede.

Al mattino ero abbastanza eccitato. Alla fine del secondo giorno ero molto eccitato. Fu a quel punto che iniziò lo stress.

Prima di accettare l’offerta della Transmeta ne parlai con diverse persone. Quando si sparse la voce che stavo considerando un’offerta di lavoro, ne ricevetti molte altre. In Finlandia ebbi un’offerta dalla Tele, che stava iniziando a usare Linux. Tramite Maddog ebbi un’offerta dalla Digital. (Niente di personale, ma Boston in inverno non è molto meglio di Helsinki. Be’, forse sì...) Parlai con delle persone di Red Hat. Mi offrirono un lavoro e mi dissero che mi avrebbero pagato meglio di Transmeta, anche se non avevano idea dello stipendio che mi era stato offerto per il semplice fatto che non ne avevamo ancora discusso. Quelli della Red Hat dissero che avrebbero superato la Transmeta anche in termini di stock option, a prescindere da quante me ne avrebbero offerte. Ma a me non interessava lavorare per una società Linux, nemmeno per una tanto fortunata da trovarsi nel mezzo del North Carolina.

Alla fine ebbi cinque offerte di lavoro senza nemmeno mettermi a cercarne uno. Quella della Transmeta era decisamente la più eccitante.

Dissi di sì. Mi fece uno strano effetto. Poi dissi all’università che me ne sarei andato. Fu allora che iniziò davvero lo stress. Per me era un grande passo, non avrei potuto tornare indietro. Aspettavamo un bambino, ci trasferivamo in un altro paese e io stavo lasciando il nido sicuro dell’università di Helsinki. Ma prima avrei dovuto scrivere la mia tesi. Col senno di poi penso che tutti quei cambiamenti insieme siano stati una buona idea. Ma era anche una follia.

Non ci fu un annuncio formale (perché avrebbe dovuto esserci?). Semplicemente la voce girò su internet, insieme al dibattito di cui abbiamo già parlato: sarei riuscito a restare fedele a Linux e al free software nell’ambiente malvagio della libera impresa, tra un cambio di pannolini e una poppata? All’epoca la gente vedeva Linux come qualcosa che era stato sviluppato soprattutto da studenti universitari, ragazzi indipendenti. Credo che il loro nervosismo fosse comprensibile.

Scrissi la mia tesi durante un lungo weekend e la consegnai qualche minuto prima di portare Tove all'ospedale dove quaranta ore dopo avrebbe dato alla luce Patricia. Era il 5 dicembre 1996. Diventare padre mi sembrò la cosa più naturale del mondo.

Nelle settimane che seguirono fui tutto preso da Patricia e dalla costante preoccupazione di ottenere le approvazioni per i nostri visti statunitensi, che non arrivavano mai. Pensammo che se fossimo stati sposati le cose sarebbero state più facili, così a gennaio - la data esatta devo sempre chiederla a Tove - andammo in comune per ufficializzare la nostra unione. C'erano tre invitati: i genitori di Tove e mia madre (mio padre era a Mosca). Fu uno strano periodo. A un certo punto spedimmo la maggior parte delle nostre cose negli Stati Uniti senza sapere quando saremmo riusciti a partire anche noi. Per salutare tutti i nostri amici tenemmo una festa d'addio alla casa, il contrario di un'inaugurazione. Venti persone si ammassarono in quel bilocale appena svuotato. Secondo le buone vecchie tradizioni finlandesi, tutti si ubriacarono.

Alla fine ci arrivarono i visti e il 17 febbraio 1997 salimmo su un volo del mattino per San Francisco. Ricordo la temperatura di Helsinki: 16°C sottozero. Ricordo la famiglia di Tove all'aeroporto. Quando li salutammo si misero a piangere. Erano molto uniti. Non ricordo se la mia famiglia c'era oppure no. Penso fossero lì. O forse no.

Atterrammo negli Stati Uniti e passammo la dogana con una bambina e due gatti. Peter Anvin era lì ad aspettarci, affittammo un'auto e andammo a Santa Clara, al condominio che avevamo scelto durante il viaggio di caccia grossa immobiliare che avevamo fatto qualche mese prima. Sembrava tutto surreale, soprattutto la differenza di 20°C rispetto alla Finlandia.

Le nostre cose non sarebbero arrivate per altri due mesi. Passammo la prima notte su un materassino gonfiabile che ci eravamo portati dietro. Il giorno dopo andammo a comprare un vero letto. Finché i nostri mobili non arrivarono in California, Patricia dovette dormire nella sua carrozzina. Era una cosa che dava molto fastidio a Tove, anche se David - ripensando ai miei primi mesi passati in un cesto della biancheria - mi ha fatto notare che era un po'una cosa ciclica. Non cucinavamo molto (non lo facciamo nemmeno ora) e non sapevamo dove andare a mangiare. Consumavamo la maggior parte dei nostri pasti nella tavola calda di un centro commerciale della zona o in un fast food. Ricordo di avere detto a Tove che dovevamo trovare qualche altro posto per mangiare.

Tra il trasloco e il fatto che mi dovevo abituare al nuovo lavoro alla Transmeta, non ebbi molto tempo da dedicare a Linux per i primi due mesi. Il nuovo lavoro occupava la maggior parte del mio tempo e le ore libere le passavo con Tove e Patricia a imparare a conoscere la nostra nuova zona. Fu un periodo molto intenso. Non avevamo un soldo. Prendevo uno stipendio da favola, ma tutti i soldi se ne andavano in mobili. Comprare un'auto fu un problema perché non avevamo modo di fornire delle credenziali di credito. Fu difficile anche dimostrare che eravamo in grado di pagare i servizi telefonici.

Il mio computer si trovava su una nave che stava superando lentamente il Capo di Buona Speranza. Fu il primo periodo in cui non feci sentire la mia voce in rete e la mia assenza preoccupò un sacco di gente. Cose del tipo: *Ok, adesso lavora per una*

società commerciale...

Molti mi chiesero senza troppi giri di parole: «Questo vuol dire che Linux non sarà più un sistema *free* ?» Spiegai che mi ero accordato con la Transmeta per poter continuare a occuparmi di Linux. E che non me ne sarei andato. (Non riuscivo a trovare un modo per dire che stavo solo riprendendo fiato.)

La vita a Transmetalandia

Uno dei problemi che avevo nello spiegare alla gente che il trasferimento negli Stati Uniti e quello nel mondo del business non mi avrebbe cambiato consisteva nel fatto che la Transmeta era una delle aziende più riservate sulla scena. C'era solo una regola a proposito di ciò di cui si poteva parlare, ed era una regola molto semplice: «Non dire nulla». E questo a volte faceva in modo che la gente di Linux si chiedesse a che razza di strana setta mi fossi unito e se ne sarei mai tornato indietro. Non potevo dire nemmeno a mia madre di cosa mi stavo occupando. Non che la cosa le interessasse.

Quello che facevo alla Transmeta non era poi così strano. La prima cosa che finii per fare fu trovare una soluzione ad alcuni problemi Linux. L'azienda stava usando molte macchine multi-processore con Linux. Io non avevo mai lavorato personalmente sui problemi SMP (Symmetric MultiProcessing) di Linux e si scoprì che molte cose non funzionavano come avrebbero dovuto. Lo presi come un affronto personale, e naturalmente dovevo sistemare la cosa.

Ma il mio vero lavoro consisteva nel far parte della squadra di softball della Transmeta.

Ah, volevo dire software. Non è che giocassimo così tanto a softball. Nessun campionato della Silicon Valley ci avrebbe accettati a meno che a nostra volta non accettassimo di dire loro su cosa stavamo lavorando.

Non so quanto la gente sappia della Transmeta. Mentre scrivo queste righe probabilmente siamo nel periodo di silenzio stampa che precede la IPO[7] (e qui incrociamo le dita...) e non siamo più un segreto per nessuno, anche se siamo tornati in «modalità riservata» a causa delle regole del SEC[8] sulle offerte pubbliche di vendita. Speriamo che per quando questo libro sarà uscito tutti avranno sentito parlare della Transmeta e avranno comprato *(messaggio sublimale:* AZIONI) una o più delle nostre CPU. Perché è questo che fa la Transmeta: CPU. Hardware.

Ma la Transmeta non fa solo hardware. Il che non è male, perché sinceramente io non saprei distinguere un transistor da un diodo nemmeno se me ne trovassi uno nel letto. La Transmeta fa hardware *semplice* che si basa su software intelligente per fare in modo che una CPU semplice sembri molto meglio di quanto non sia in realtà: come un x86 standard Intel compatibile, in effetti. E con il fatto che l'hardware diviene più piccolo e più semplice, la CPU finisce con l'avere meno transistor, il che a sua volta significa che usa meno energia, che come tutti sanno sta diventando una questione sempre più importante in un mondo all'insegna della mobilità. E a causa di questo software intelligente che la Transmeta ha una squadra di software abbastanza

---

[7] Offerta pubblica di vendita [n.d.t]

[8] Il corrispettivo americano della Consob [n.d.t.]

grande, ed è per questo che io mi sono trovato lì.

Tutto ciò mi si adattava alla perfezione. Un'azienda non-Linux che faceva qualcosa di tecnicamente interessante (e questo è l'*understatement dell'anno:* non so ancora oggi di un'azienda che abbia mai provato seriamente a fare ciò che la Transmeta), per di più in un'area che conoscevo bene: la programmazione a basso livello della tutto sommato esoterica famiglia di CPU 80x86. Come certamente ricorderete, era stato proprio il fatto di imparare a conoscere quel tipo di CPU, montata dal mio primo personal computer, che aveva dato il via a tutto il progetto Linux.

Per me era importante anche il fatto che la Transmeta non fosse un'azienda Linux. Non fraintendetemi: adoravo risolvere i problemi Linux alla Transmeta, e sono stato coinvolto in diversi progetti interni che riguardavano Linux. (In effetti oggi è probabilmente impossibile trovare una società tecnologica seria che non abbia questo tipo di progetti.) Ma alla Transmeta Linux era secondario, il che era esattamente ciò che volevo. Potevo continuare a occuparmi di Linux, ma non sentivo di dover fare dei compromessi tecnici che favorissero gli obiettivi aziendali rispetto a Linux stesso. Potevo continuare a pensare a Linux come a un hobby in cui le mie decisioni erano determinate solo ed esclusivamente da considerazioni di carattere tecnologico.

Così durante il giorno lavoravo per la Transmeta. Scrivevo e gestivo l'«interprete x86» che utilizziamo ancora oggi (anche se adesso è qualcun altro a gestirlo). L'interprete è fondamentalmente il pezzo di software Transmeta che legge le istruzioni Intel una per volta e le esegue (ovvero «interpreta» il linguaggio dell'architettura 80x86, un'istruzione alla volta). In seguito finii per fare altre cose, ma fu così che entrai nello strano e affascinante mondo dell'emulazione hardware.

Di notte dormivo.

Il mio accordo con la Transmeta era chiaro: mi erano state date delle vaghe assicurazioni che avrei potuto occuparmi di Linux anche durante le ore lavorative. Credetemi: ne ho decisamente approfittato.

Molte persone credono nelle giornate lavorative lunghe e nei doppi, tripli o addirittura quadrupli turni. Io non sono tra queste. Né la Transmeta né Linux mi hanno mai impedito di farmi un bel sonno. In effetti, se volete sapere tutta la verità, credo fermamente nel sonno. Alcuni pensano che questo voglia dire essere pigri: io li prenderei a cuscinate in faccia. Ho una scusa perfetta e mi ci atterrò scrupolosamente: magari dormendo puoi perdere qualche ora del tuo tempo produttivo, diciamo dieci ore al giorno, ma in quelle poche ore in cui sei sveglio lo sei per davvero e il o cervello funziona a tutta birra. Quasi sempre.

IV.

Benvenuti nella Silicon Valley. Una delle prime cose che dovetti fare appena atterrato in questa strana galassia fu incontrare le star.

Ricevetti una e-mail dalla segretaria di Steve Jobs in cui mi diceva che se avevo un paio d'ore libere gli sarebbe piaciuto incontrarmi. Non sapendo di cosa si trattasse,

dissi che non c’era problema.

L’appuntamento era alla sede centrale della Apple in Infinite Loop Drive. Dovevo vedere Jobs e il suo responsabile tecnico, Avie Tevanian. Erano i tempi in cui la Apple stava iniziando a lavorare sull’OS X, il sistema operativo basato su Unix rilasciato al pubblico solo nel settembre 2000. Non ci furono molte formalità in quell’incontro. Fondamentalmente Jobs iniziò col dirmi che sulla scacchiera c’erano solo due giocatori, Microsoft e Apple, e che pensava che la cosa migliore che potessi fare per Linux fosse andare a letto con Apple e cercare di far lavorare la gente dell’open source su Mac OS X.

Io non me ne andai subito solo perché volevo sapere qualcosa sul nuovo sistema operativo. É basato su Mach, il microkernel sviluppato alla Carnegie Mellon University. A metà degli anni Novanta sembrava che il Mach dovesse essere il sistema operativo definitivo e un sacco di gente se e interessava. In effetti IBM e Apple utilizzarono Mach come base per Taligent, il sistema operativo destinato al fallimento che svilupparono in joint-venture.

Jobs sottolineò con enfasi il fatto che il kernel di basso livello di Mach era *open source*. Fu come svelarmi la fregatura di tutta la faccenda: cosa importa se il sistema operativo di base, la roba che sta al cuore di tutto, è *open source* quando sopra a questo si trova il livello Mac, che non lo è?

Non poteva sapere che personalmente non avevo un’altissima opinione di Mach. Per essere più precisi lo definirei una vera e propria schifezza. Contiene tutti gli errori di progettazione possibili e immaginabili, ed è riuscito addirittura a inventarne di inediti. Una delle argomentazioni contro il microkernel è sempre stata quella delle prestazioni. Così un sacco di gente ha fatto delle ricerche per capire come trasformare il microkernel in qualcosa che avesse delle buone prestazioni. Tutti i risultati di questa ricerca sono confluiti in Mach, che è quindi diventato un sistema estremamente complesso, con regole proprie. E ciononostante non ha affatto delle prestazioni eccellenti.

Avie Tevanian era stato uno dei responsabili di Mach quando era ancora un progetto universitario. Era abbastanza interessante discutere quelli che lui e Steve ritenevano i problemi più importanti. Ma da un punto di vista tecnico eravamo in quasi totale disaccordo. Io non pensavo affatto che ci fossero dei motivi per cui la gente dell’open source e di Linux dovesse essere coinvolta nella cosa. Certo, potevo capire perché loro volessero più sviluppatori open source nel proprio sistema: vedevano l’abbrivio che stava prendendo Linux. Ma non credo che ne avessero un’idea corretta. Non credo che Jobs capisse che Linux avrebbe potuto avere più utenti di Apple, anche se si trattava di una base di utenti molto differente. E non credo che Steve oggi liquiderebbe Linux come sistema desktop con la stessa facilità con cui lo fece tre anni fa.

Spiegai che Mach non mi piaceva granché. Per ragioni ben comprensibili la cosa non creò una buona atmosfera. Avevano certamente già sentito prima le mie argomentazioni. Naturalmente io ero molto fermo su Linux e Tevanian su Mach. Era interessante vedere come discutevamo alcune questioni tecniche. Uno dei primi problemi che sollevai riguardava il modo in cui avevano progettato di supportare le vecchie applicazioni Mac con il nuovo sistema operativo. Volevano fare tutte le

vecchie cose con un livello di compatibilità. Tutte le vecchie applicazioni Mac avrebbero girato dentro un nuovo processo appiccicato ad hoc. Ma uno dei limiti principali del vecchio Mac è la mancanza di una protezione di memoria, e questa soluzione non fa nulla per eliminare il problema. Solo le nuove applicazioni Mac avrebbero avuto la protezione di memoria. Non mi sembrava una cosa sensata.

C'erano delle divergenze di base anche nelle nostre visioni del mondo. Steve era Steve, esattamente come viene ritratto dalla stampa. Gli interessavano i suoi obiettivi, e in particolare l'aspetto marketing. A me interessava l'aspetto tecnico e non importava granché né dei suoi obiettivi né delle sue argomentazioni. La sua argomentazione principale era questa: se volevo arrivare al mercato desktop dovevo unire le mie forze con quelle della Apple. La mia reazione era: perché mai? Perché dovrebbe interessarmi la storia della Apple? Non pensavo che ci fosse alcunché di interessante nella Apple. E il mio scopo nella vita non era quello di conquistare il mercato desktop. (Certo, accadrà, ma non è mai stato quello il mio obiettivo.)

Non stava molto ad argomentare. In sostanza dava per scontato che la cosa mi interessasse. Non riusciva proprio a immaginare che ci fossero interi segmenti della razza umana il cui ultimo pensiero era l'aumento della quota di mercato del Mac. Credo fosse davvero sorpreso da quanto poco mi importassero le enormi dimensioni del mercato del Mac o della Microsoft. E non posso certo incolparlo di non avere saputo in anticipo quanto poco mi piacesse Mach.

Ma anche se mi trovai in disaccordo con quasi tutto quello che diceva, quell'uomo mi piacque.

Poi ci fu la prima volta che incontrai Bill Joy. Lo piantai in asso.

Ok, se devo essere sincero non avevo capito chi fosse, la prima volta che lo incontrai. Avvenne a una presentazione di Jini. Jini è il linguaggio per agenti interattivi della Sun Microsystems, un'estensione di Java. Serve a creare delle reti invisibili tra sistemi completamente differenti. Se avevi una stampante in Jini, tutto ciò che nella stessa rete parlava Jini poteva usarla automaticamente.

La Sun Microsystems aveva invitato me e una dozzina di altre persone dell'open source e tecnici vari a un'anteprima privata che avrebbe avuto luogo in una stanza d'albergo nel centro di San José, durante Java World. La ragione per cui ci avevano invitati? Stavano realizzando Jini in quello che alla Sun Microsystems passava per *open source.*

Quando ci andai, sapevo che Bill Joy era lì. Era stato l'uomo chiave dietro BSD Unix e in seguito era entrato alla Sun come capo della ricerca scientifica. Non lo avevo mai incontrato prima. Lui venne da me, mi disse che era Bill Joy e io non ebbi alcuna reazione. Non ero venuto lì per incontrarlo ma per vedere ciò che la Sun pensava *dell'open source* e come sarebbero entrati in quel giro. Alcuni minuti più tardi Bill in persona stava spiegando le ragioni per realizzare quel progetto in *open source.* Vi fu anche una dimostrazione limitata del sistema.

Poi iniziarono a spiegare le loro licenze. Fu orribile. Una cosa stupida. Fondamentalmente tutto si riduceva al fatto che se qualcun altro voleva usare il sistema per scopi anche solo limitatamente commerciali, non sarebbe stato affatto *open source.* Mi sembrò un'idea totalmente idiota. Ero davvero scocciato dal fatto che sull'invito avessero tanto sbandierato quella faccenda *dell'open source.* Era *open*

source nel senso che potevi leggere le sorgenti, ma se volevi fare delle modifiche o integrarlo nella tua infrastruttura, dovevi comprare la licenza dalla Sun. Se qualcuno alla Red Hat avesse voluto rendere l'ultima versione CD di Linux Red Hat compatibile con Jini, l'azienda avrebbe dovuto acquistare la licenza Jini da Sun.

Posi alcune domande per vedere se avevo capito tutto bene.

Poi li piantai in asso.

Mi faceva tanto incazzare il fatto che avessero portato lì delle persone con il trucco *dell'open source* che quando scoprii di cosa si trattava dissi testualmente: «Lasciate stare, non mi interes-sa» e me ne andai.

Avevo capito che mi volevano lì solo per darmi delle informazioni e che al limite - se fossi stato entusiasta del progetto - mi avrebbero chiesto delle dichiarazioni per la stampa o qualcosa del genere. Il loro piano gli si rivoltò contro. Ma forse così impareranno. Pare che in seguito qualcuno li abbia convinti a rendere *open source* il loro Star Office.

Quindi forse è solo che c'è un tempo per ogni cosa.

Mi dicono che quell'incontro proseguì fino alla cena e che nessun altro se ne andò.

La seconda volta che incontrai Bill Joy andò molto meglio. Più o meno un anno e mezzo dopo mi invitò a mangiare un sushi.

La sua segretaria mi telefonò per concordare una data. Bill vive e lavora in Colorado, ma passa una settimana al mese nella Silicon Valley. Andammo da Fuki Sushi a Palo Alto. É uno dei migliori posti in cui mangiare sushi nella Valley. Naturalmente non è nulla in confronto a Blowfish Sushi di San Francisco con la sua continua animazione giapponese, o Tokyo Go Go alla Mission con la sua clientela alla moda, o Sushi Ran di Sausalito con la sua gestione aristocratica, o Seto Sushi di Sunnyvale che ha il migliore sushi di tonno piccante di tutti.

Ok, allora eravamo da Fuki Sushi e fu abbastanza divertente perché Bill cercava di avere del vero wasabi. Non lo sapevo all'epoca, ma nella maggior parte dei ristoranti giapponesi negli Stati Uniti quello che passa per wasabi in realtà è solo rafano colorato. Ho scoperto che la pianta del wasabi cresce solo nei fiumi giapponesi ed è difficile da coltivare a livello industriale. Bill cercava di spiegare tutto questo alla cameriera e lei non capiva il concetto. Era giapponese, ma pensava che il wasabi fosse wasabi. Lui le chiese di parlare con i cuochi.

Questo tira e molla fu abbastanza divertente. Era una cena informale. Praticamente mi aveva già chiarito che se avessi voluto lavorare per la Sun mi sarebbe bastato dirglielo e lui avrebbe fatto in modo che succedesse qualcosa. Ma non era quella la questione principale. Era più un'opportunità di discutere alcuni argomenti. Iniziò a ricordare di quando aveva gestito BSD Unix per cinque anni e di come era arrivato ad apprezzare il fatto di avere un supporto commerciale tramite la Sun. Mi parlò di quanto fosse importante avere il tipo di supporto commerciale che un'azienda come la Sun può fornire. Fu divertente e interessante sentirlo parlare dei primi passi di Unix. Per me non fece la minima differenza il fatto che non riuscimmo a farci servire del vero wasabi. Ricordo perfettamente di avere pensato che Bill era probabilmente il più simpatico e interessante tra i pezzi da novanta che avevo incontrato nella Silicon Valley.

Facciamo un salto in avanti di tre anni. Sto sfogliando la rivista «Wired» e mi

imbatto in un suo articolo orribilmente negativo sulla tecnologia intitolato *Il futuro non ha bisogno di noi.* Sono un po'deluso. Certo che il futuro non ha bisogno di noi! Ma Bill non doveva essere per forza tanto negativo sull'argomento.

Non voglio attaccare quell'articolo riga per riga, ma in generale ritengo che la cosa più triste che ci possa accadere consisterebbe nel continuare ad andare sempre avanti, senza in realtà evolverci. Bill sembrava ritenere che i progressi in campi come l'ingegneria genetica ci fanno perdere la nostra umanità. Tutti pensano sempre che qualcosa di diverso sia inumano perché ora *siamo* umani. Ma continuando a evolverci, qualsiasi cosa accada, tra 10.000 anni non saremo più umani secondo gli standard attuali. Avremo una forma d'umanità differente.

Bill, in quell'articolo, sembrava spaventato da tutto questo. La mia sensazione è che sia innaturale - e inutile - cercare di rallentare l'evoluzione. Invece di far accoppiare due razze diverse di cani per ottenerne una terza, ci rivolgeremo alla genetica; mi sembra inevitabile che questo accada anche per le persone. Secondo me cambiare la razza umana attraverso la genetica è meglio che mantenere lo status quo. Credo che in linea generale sarebbe molto più interessante assicurarci che vi sia un'evoluzione non solo degli uomini, ma anche della società, in qualsiasi direzione essa vada. Non si può fermare la tecnologia e non si possono fermare i progressi della conoscenza sul funzionamento dell'universo e sulla progettazione degli esseri umani. Si sta muovendo tutto tanto in fretta che alcune persone, come Bill Joy, ne sono spaventate. Ma io la vedo come una parte della nostra evoluzione naturale.

Non sono d'accordo con Bill sul modo in cui dovremmo affrontare il futuro, come non ero d'accordo con la sua idea di *open source.* Non ero d'accordo nemmeno con Steve obs sulla sua idea di tecnologia. Sembrerebbe proprio che abbia passato il mio primo anno nella Silicon Valley a non essere d'accordo con qualcuno, ma non è vero. Ho scritto un sacco di codice e ho portato Patricia allo zoo e in generale ho ampliato i miei orizzonti. Imparando la terribile verità sul wasabi, per esempio.

V.

Il nostro successo dalla sera alla mattina

Leggete mai i newsgroup di sostegno? Lo scopo della loro esistenza è esclusivamente quello di sostenere qualcosa, il che significa inevitabilmente dare addosso a qualcos'altro. Così se andate a visitarli trovate solo nonsense del tipo «Il mio sistema è meglio del tuo. È una specie di masturbazione online.

Vi sto parlando dei newsgroup di sostegno, nonostante la loro insensatezza, perché in effetti vi permettono di farvi un'idea di cosa sta succedendo. Così quando le aziende decisero che Linux era il sistema operativo da coccolare, la crescita del supporto commerciale non venne discussa inizialmente sulla stampa o nei punti vendita, ma sui newsgroup di sostegno.

Facciamo un passo indietro. Nella primavera del 1998 una terza bionda era entrata nella mia vita: Daniela Yolanda Torvalds nacque il 16 aprile. Era la prima Torvalds a essere cittadina americana a tutti gli effetti. Tra lei e Patricia c'erano sedici mesi,

come tra Sara e me. Ma sono sicuro che non si daranno battaglia come me e mia sorella, certamente non con l'influenza moderatrice di Tove (per non parlare delle sue capacità di karateka).

Due settimane prima della nascita di Daniela la comunità *open source* - che fino a poco tempo prima veniva chiamata «comunità *free software*» - fece il passo in avanti più grande della sua storia quando Netscape aprì il codice sorgente della sua tecnologia browser in un progetto chiamato Mozilla. Da una parte questa notizia mise in fermento tutti i frequentatori dei newsgroup, perché aumentava la visibilità *dell'ope*n *source.* Ma molte persone, me compreso, erano abbastanza preoccupate. Netscape all'epoca era nei guai, grazie soprattutto a Microsoft, e l'apertura del suo browser sembrava una mossa disperata. (È buffo se si pensa che le radici del browser Netscape affondavano *nell'open source.* Era nato infatti come progetto della University of Illinois.)

La gente sui newsgroup esprimeva i propri timori che Netscape avrebbe incasinato le cose e rovinato la reputazione *dell'open source.* Ora ci sarebbero stati due grossi progetti *open source* in circolazione - Netscape e Linux - e il ragionamento di base era che se Netscape, il più noto dei due, fosse fallito, ne avrebbe sofferto anche la reputazione di Linux.

E in effetti Netscape fallì. Almeno in buona parte. L'azienda aveva dei problemi a coinvolgere nel progetto gli sviluppatori *open source* sul lungo periodo. Si trattava di un ammasso di codice e gli unici che riuscivano a orientarsi al suo interno erano quelli della Netscape.

Il progetto non aveva il destino segnato solo a causa delle sue dimensioni. Il problema era anche il fatto che Netscape non riusciva a diffondere in *open source* tutto quanto, ma solo la versione di sviluppo, che quando venne rilasciata dall'azienda era abbastanza disastrata. Netscape non poteva assegnare una licenza GPL al browser, perché il codice non era tutto suo - la parte Java era in licenza dalla Sun, per esempio. Non tutti nel newsgroup erano d'accordo con la licenza Netscape. In generale era una licenza abbastanza lasca, ma quando ti chiami Richard Stallman non ti possono piacere le cose lasche.

Pensai fosse meraviglioso che Netscape avesse fatto questo passo, ma non lo vedevo come un successo personale. Ricordo che Eric Raymond la prese in modo molto personale. Ne era felicissimo. Il suo saggio *La cattedrale e il bazar,* che spiegava alla perfezione la filosofia e la storia *dell'open source,* era uscito un anno prima e veniva citato come una delle motivazioni alla base della decisione di Netscape. Raymond era un attivista *dell'open source.* Era stato alla Netscape un sacco di volte per cercare di convincerli ad aprire il loro browser. Io ci andai una volta sola. In effetti Eric era stato a visitare tutta una serie di aziende per diffondere la novella *dell'open source.* A me interessava la tecnologia, non il proselitismo.

Nel giro di 24 ore dal rilascio di Mozilla, un gruppo australiano che si chiamava Mozilla Cripto Group aveva già creato il modulo di crittografia. All'epoca chi non era cittadino americano non poteva utilizzare sistemi crittografici creati sul suolo statunitense. All'improvviso qualcuno in Australia aveva creato un'alternativa utilizzabile da tutti. Ma c'era un problema. A causa delle leggi sull'esportazione dell'epoca, il progetto Mozilla non poteva integrare codice australiano. E se questo

fosse stato trasferito negli Stati Uniti, non avrebbe più potuto uscirne. E questo significava che uno dei primi successi del grande esperimento di Netscape non poteva entrare a far parte di Mozilla.

Eravamo tutti preoccupati perché Netscape aveva ricevuto una forte copertura mediatica. E in quel primo anno tutti ci andarono con i piedi di piombo. Nessuno voleva dire qualcosa di negativo su Netscape per paura che poi la stampa parlasse male dell'open source e spaventasse delle aziende potenzialmente interessate alla cosa.

Ma due mesi dopo la mossa di Netscape la Sun Microsystems entrò in gioco dichiarando che sarebbe stato il primo grande venditore di hardware a unirsi a Linux International. Avrebbe supportato Linux sui propri server. L'azienda che aveva escogitato il pessimo trucco della licenza di Jini aveva deciso che Linux andava preso sul serio. I newsgroup furono inondati da autocongratulazioni. Con la Sun a bordo, gli sviluppi di Linux passarono dalle discussioni su internet alla stampa specializzata. Iniziavano a essere interessati alla cosa anche gli outsider, sebbene per lo più si trattasse di tecnici.

Poi arrivò la IBM.

La IBM era nota per la propria stolidità, per cui tutti vennero presi di sorpresa quando a giugno l'azienda annunciò che avrebbe venduto e supportato Apache, la più popolare tra le versioni commerciali di Linux utilizzate per i server Web. Apache poteva girare su AIX, lo UNIX di IBM, ed era probabilmente ciò che faceva un sacco di gente che aveva acquistato prodotti IBM. Fu così che Apache attrasse l'attenzione dell'azienda. Qualcuno doveva aver notato che la maggior parte di quei server finivano per utilizzare Apache, così pensarono che avrebbero venduto più server se avessero avuto l'expertise interna per dare assistenza a questi clienti. O forse erano stati spinti da clienti che avevano detto loro che avrebbero comprato server IBM ma vi avrebbero fatto girare sopra Apache.

Installare Linux su un computer è relativamente semplice. Ma per molte aziende uno dei grandi problemi era storicamente questo: a chi diamo la colpa se qualcosa va male? Naturalmente ci sono aziende Linux come Red Hat che offrono un servizio di assistenza, ma per i clienti era un vantaggio psicologico sapere che avrebbero avuto IBM a coprire loro le spalle. Quando IBM ini-ziò a entrare nell'open source molti sospettarono che si trattasse solo di una mossa pubblicitaria. Ma alla fine si capì che non era così. IBM fece una prima prova installando e supportando Linux sui propri server più grossi. Poi si tuffò a capofitto nel progetto. Fu la volta dei server PC più piccoli. Poi toccò ai PC veri e propri. Poi ai portatili. L'azienda ha annunciato che quest'anno investirà un miliardo di dollari su Linux.

IBM faceva la maggior parte del lavoro su Linux per conto proprio. Credo che una delle ragioni per cui apprezzavano Linux era che potevano fare ciò che volevano senza dover badare a problemi di licenza. Non dimentichiamo che IBM ha avuto la sua bella fetta di rotture di scatole in questo ambito. È stata fregata da Microsoft dopo che avevano sviluppato insieme il sistema operativo OS/2, che alla fine si rivelò solo una versione anabolizzata di Windows. Microsoft non supportò OS/2 perché non era interessata a dividere il proprio mercato. Per loro da quel progetto uscì Windows NT. Ma OS/2 non ripagò mai la IBM dei miliardi di dollari che vi avevano investito. E poi

IBM era ossessionata dai problemi di licenza con Java. Credo fossero felicissimi di non avere quel tipo di problemi con Linux.

Non c'è dubbio che quello di IBM fu il colpo grosso di Linux. E sui newsgroup creò solo una grande eccitazione, e non quella specie di paranoia provocata dall'annuncio di Netscape o l'anticommercialismo strisciante che ogni tanto (Ok, spesso) divide i seguaci di Linux.

A luglio Informix annunciò che avrebbe portato i propri database in Linux, il che voleva dire che potevi far girare i database Informix anche se il tuo computer lavorava in Linux. All'epoca non è che fosse un grande affare. L'azienda stava attraversando dei problemi finanziari, ma era comunque uno dei tre principali produttori di database. Il popolo di Linux fu abbastanza felice di questo sviluppo e si mise a scrivere saggi di autocongratulazione nei gruppi di sostegno Linux.

Poche settimane dopo entrò in gioco all'improvviso la Oracle. La Oracle dominava il mercato dei database. Molto tempo prima dell'annuncio erano girate delle voci (sui newsgroup) sul fatto che l'azienda stesse lavorando internamente per portare delle cose in Linux. E dato che Oracle è sinonimo di server Unix, il passo verso Linux non era lunghissimo. Ma se seguivi i newsgroup capivi che ormai eravamo definitivamente esplosi. L'annuncio della Oracle ebbe un fortissimo impatto psicologico, anche se quello tecnologico era pari a zero.

Come era accaduto in precedenza per l'annuncio di IBM, la mossa della Oracle non fu recepita solo dalla comunità Linux, ma anche dalla dirigenza delle aziende, da quelli che vengono solitamente chiamati decision maker. Non avrebbero più potuto dire che non adottavano Linux perché le loro aziende facevano un forte utilizzo di database.

Per quanto gratificanti, queste notizie non mi cambiarono la vita. Io e Tove ci stavamo spupazzando due bambine adorabili. La maggior parte delle ore che non dedicavo alla famiglia le passavo a gestire Linux, sia a casa sia in ufficio. Per non favorire un'unica versione del sistema operativo, usavo Red Hat al lavoro e SuSE, una versione europea, a casa. A un certo punto decisi che non facevo abbastanza esercizio fisico, così mi misi in testa di fare in bicicletta le sei miglia tra casa nostra e la sede della Transmeta. Era lunedì. Non c'erano salite da affrontare, ma un vento molto forte mi soffiava contro, rendendo il tutto più difficile di quanto avessi desiderato. Quando uscii dal lavoro dieci ore più tardi il vento era cambiato e naturalmente anche questa volta mi soffiava dritto in faccia. Telefonai a Tove e le chiesi di venirmi a prendere. Non c'è bisogno che vi dica che non andai mai più a lavorare in bicicletta.

Vi ho raccontato questo innocuo dettaglio per mostrarvi come gli sviluppi di Linux non avessero conseguenze sulla mia vita quotidiana. La maggior parte dell'attività si svolgeva dentro le aziende. I tecnici che conoscevano Linux da un pezzo venivano contattati dai dirigenti delle loro aziende che avevano letto degli articoli su Linux o ne avevano sentito parlare. Chiedevano ai loro tecnici cosa fosse tutta quella storia. Poi, una volta capiti i vantaggi, decidevano di far girare Linux sui propri server.

La situazione si stava evolvendo in tutti i dipartimenti informatici del mondo, anche se l'epicentro erano indubbiamente gli Stati Uniti. Si trattava raramente di una decisione basata sulla gratuità di Linux, perché il software in sé rappresenta di fatto

una minima parte di un investimento del genere. Il servizio e l'assistenza sono molto più costosi. Ad attrarre i dirigenti era tendenzialmente la semplicità delle argomentazioni tecniche: Linux era più forte della concorrenza, costituita da Windows NT e dalle numerose varianti di Unix. Ma soprattutto la gente detestava dover fare le cose in base a come le aveva decise Microsoft (o chiunque altro). Con Linux si possono fare cose che non si possono fare con i sistemi della concorrenza. I primi a usare Linux lo fecero perché così potevano accedere al codice sorgente in un modo impossibile con il software commerciale.

Da quel punto di vista le cose non erano molto cambiate da quando avevo rilasciato la versione 0.01 dalla mia camera da letto. Linux era più flessibile degli altri sistemi. Eri tu a prendere le decisioni. E, almeno nel caso dei server Web, non conteneva le «gonfiature» - tutte le caratteristiche inutili - che appesantivano i sistemi operativi della concorrenza.

Linux aveva un altro elemento a suo favore: nonostante la sua crescente popolarità come sistema operativo per server Web, non occupava affatto una nicchia. È importante per capire il successo di Linux.

I computer mainframe erano una nicchia. Unix in generale occupava una serie di nicchie (quella dei supercomputer del ministero statunitense della difesa, quella delle banche). Chi vendeva i sistemi operativi per i mainframe o per altri grandi sistemi guadagnava molto perché li faceva pagare carissimi. Poi arrivò la Microsoft e iniziò a venderli a novanta dollari. Microsoft non cercò di conquistare la nicchia delle banche o altro, ma all'improvviso si trovò dappertutto. Era come essere invasi dalle locuste. É difficile liberarsi da un'invasione del genere. (Non che abbia qualcosa contro le locuste, mi piacciono tutti gli animali.)

È molto meglio essere ovunque e conquistare tutte le nicchie. Fu ciò che fece la Microsoft. Pensate a un organismo fluido che scorre ovunque trovi uno spiraglio. Se viene persa una nicchia, non è un grosso problema. L'organismo avvolge il mondo, scorrendo ovunque trovi dell'interesse.

La stessa cosa sta succedendo oggi a Linux. Scorre ovunque vi sia dell'interesse. Linux non occupa una sola nicchia. È piccolo e flessibile e trova il modo per entrare quasi dappertutto. Lo trovate nei supercomputer, in posti importanti come i Fermilabs del governo americano o alla NASA. Ma lì arriva scorrendo fuori dallo spazio server. Dove a sua volta arriva scorrendo fuori dallo spazio desktop, che è il punto da cui io sono partito. Allo stesso tempo troverete Linux nascosto in strumentazioni che vanno dai freni antiblocco agli orologi.

Guardatelo scorrere.

Ma abbiamo anche un altro vantaggio. Gli esponenti migliori e più brillanti della prossima generazione usano il tuo prodotto per una questione di passione generazionale. Per la generazione precedente non era stato così. Non erano Microsoft o il DOS ad appassionare la gente, quanto i PC. Se eri appassionato di PC, lo eri anche di DOS. Non c'era molta scelta.

E questo è stato un grande vantaggio per la diffusione di Microsoft.

Se guardate i ragazzini più brillanti in circolazione, non usano tutti Linux, ma la maggior parte sì. Certo, uno dei motivi per cui la filosofia *open source* e Linux hanno un grande seguito nelle università è semplice: il sentimento anti-establishment. (Lo

stesso sentimento anti-establishment che ha avuto tanto impatto sulla vita di mio padre.) Una cosa del tipo: La Grande E Malvagia Microsoft & Il Maledetto Avido Schifosamente Ricco Bill Gates contro Ci-Siamo-Dentro-Per-Amore-Del-Free-Software-Per-Tutti l'(Apparentemente) Modesto Eroe Popolare Linus B. Torvalds. Questi ragazzi si laureano e vanno a lavorare nelle aziende, dove si portano dietro il proprio amore per Linux.

Così gente che si è avventurata nei meandri di Microsoft mi racconta di avere visto la mia faccia sui bersagli delle freccette. Il mio solo commento: è troppo facile colpirmi, con il naso che mi ritrovo.

Ma sto correndo troppo. Il grande annuncio di IBM nella primavera del 1998 fu seguito da altri annunci da parte di tutti i grandi produttori di hardware. Ad agosto la rivista «Forbes» «scoprì» il nostro piccolo mondo mettendo una mia fotografia in copertina con le parole «Pace, Amore, Software;». A mano a mano che le aziende si impegnavano (inevitabilmente) a lavorare su Linux, non c'era più bisogno di passare in rassegna i newsgroup di sostegno per leggere il futuro.

VI.

Linux aveva conquistato il cuore del mondo con un improbabile vincitore delle Olimpiadi proveniente da uno sconosciuto paese del terzo mondo.

Io ero il conferenziere del momento. In un'intervista Eric Raymond spiegò che parte del mio fascino (ehm...) stava nel fatto che ero «apparentemente meno strano di molti altri hacker». Ok. È l'opinione di un hacker. La situazione non piaceva a tutti. Richard Stallman fece una campagna per cambiare il nome di Linux in GNU/Linux, secondo la logica che mi ero basato sul compilatore GNU gcc e su altri strumenti e applicazioni di free software per far decollare Linux. Altri erano sempre più irritati dal fatto che Linux si stesse facendo strada nel mondo delle aziende.

La stampa giocava sulla dicotomia tra Idealisti e Pragmatici (non sono parole mie!) all'interno della comunità Linux, composta ormai da centinaia di migliaia di persone. Secondo quella suddivisione, coloro che temevano che gli ideali di Linux fossero incompatibili con gli obiettivi del capitalismo erano gli Idealisti. Io ero a capo dei Pragmatici. Ma quell'analisi mi sembrava un *nonsense* giornalistico, un tentativo semplicistico di far rientrare tutto in uno schematico universo in bianco e nero. (Ho lo stesso problema con il modo in cui certa gente vede il fenomeno Linux come una guerra Linux contro Microsoft, quando di fatto è qualcosa di totalmente differente e di molto più ampio. É un modo più organico di diffondere la tecnologia, la conoscenza, la ricchezza e di divertirsi rispetto a qualsiasi cosa il mondo del commercio abbia mai conosciuto.)

Per me era un falso problema. Senza interessi commerciali, come avrebbe potuto Linux fluire in nuovi mercati? Come avrebbe potuto creare opportunità di innovazione? Come avrebbe potuto raggiungere le persone che volevano un'alternativa - un'alternativa *free* - alla pessima tecnologia in circolazione? Quale modo più realistico per raggiungere l'affermazione *dell'open source,* se non tramite

la sponsorizzazione delle aziende? E quale modo migliore per liberarsi della parte meno interessante del lavoro, roba noiosa tipo la manutenzione e l'assistenza, rispetto a farlo fare alle aziende?

L'*open source* è basato sull'idea di permettere a chiunque di entrare in gioco. Perché le aziende, che stanno alla base di buona parte dei progressi tecnologici della società, dovrebbero esserne escluse, a patto che giochino secondo le regole? L'*open source* non può fare altro che migliorare la tecnologia creata dalle aziende, e forse le può anche rendere meno avide.

E anche se volessimo fermare le forze del commercio, cosa potremmo fare? Io non avevo intenzione di suggerire che ci nascondessimo, ci dessimo all'underground, ci rifiutassimo di parlare con le aziende.

I sentimenti anticommerciali hanno sempre fatto parte della comunità *open source,* ma è stato solo quando Linux è diventata una parola d'ordine alla moda un po'in tutto il mondo che la posta in gioco si è fatta importante. I newsgroup erano tutto un fiorire di deliri paranoici di fanatici vocianti. Nessuno degli sviluppatori di Linux con cui avevo a che fare si mostrava preoccupato. Ma c'erano altri che sbraitavano di come Red Hat e altre aziende avrebbero pervertito l'idea di *open source* e di come alcune persone stessero perdendo il proprio idealismo.

In una qualche misura è probabilmente vero che una parte del popolo *dell'open source* si stava allontanando dal proprio idealismo. Ma mentre per alcuni questa era una sconfitta, a me sembrava che ci desse semplicemente più possibilità di scelta. Per esempio i tecnici che si preoccupavano di cose tipo dar da mangiare ai propri figli ora avevano una possibilità. Potevi continuare a essere un'idealista senza macchia e senza paura oppure potevi entrare a far parte della nuova schiatta commerciale. Non ci perdi niente quando arriva qualcun altro e ti dà una nuova possibilità. Prima, evidentemente, non potevi scegliere se non di restare duro e puro.

A proposito: io non ho mai sentito di far parte della schiera degli Idealisti. Certo, ho sempre visto l'*open source* come un modo per rendere un po'migliore il mondo. Ma soprattutto lo considero un modo per divertirmi. Il che non è molto idealistico.

E ho sempre pensato che gli idealisti siano interessanti, ma anche noiosi e a volte spaventosi.

Per conservare un'opinione molto forte, devi escludere tutte le altre. Il che significa che devi diventare irragionevole. Questo è uno dei problemi che ho con la politica americana rispetto a quella europea. Nella versione americana del gioco decidi qual è il confine che ti divide dal nemico e la tua abilità sta nel mantenere salda questa divisione. I politici europei tendono a vincere dimostrando di saper sostenere la cooperazione.

Così io sono legato a un approccio conciliatorio. L'unica volta in cui mi sono davvero preoccupato di una deriva commerciale è stata agli inizi, quando Linux non si era ancora fatto un nome. A quel punto, se le aziende avessero cooptato Linux, non avrei potuto fare nulla. Ma è evidente che oggi non è più così. Una preoccupazione che serpeggiava nei newsgroup nel 1998 era che le aziende non avrebbero dato nulla in cambio. In un certo senso io mi dovevo fidare dei nuovi attori aziendali come gli sviluppatori di Linux si dovevano fidare di me. E le aziende si dimostrarono degne di questa fiducia. Non si sono mai ritirate. Fino a ora il nostro è stato un rapporto molto

positivo.

Conferenziere, proprietario del trademark di Linux, gestore del kernel di Linux: sentivo un senso di responsabilità sempre più forte. Mi sentivo sempre più responsabile del fatto che milioni di persone contassero su Linux e c'era un'immensa pressione per essere certi che funzionasse nel modo più affidabile possibile. Per me era importante aiutare le aziende a capire cosa fosse l'*open source.* Non c'era alcuna guerra in corso, per quanto mi riguardava, tra aziende avide e hacker altruisti.

No, non stavo rinunciando ai miei ideali incontrandomi con la Intel quando mi chiesero di aiutarli con il baco di blocco Pentium F0 0F (sento che vi state chiedendo: «baco di blocco Pentium F0 0F?» Lo so, noi informatici ci inventiamo dei nomi parecchio strambi. «F0 0F» è la rappresentazione esadecimale dei primi due byte di una sequenza di istruzione illecita che faceva bloccare la CPU dei Pentium. Di qui il nome.) No, non era ipocrita per me promuovere le meraviglie di un codice *open source* e al contempo intascare lo stipendio di un'azienda tanto chiusa da non dire nemmeno in giro di cosa si occupasse. Il fatto era che io ammiravo - e lo ammiro tutt'oggi - il chip a bassa alimentazione che la Transmeta stava sviluppando, e lo vedevo come il progetto tecnologico più interessante sul mercato (nonché uno dei progetti dalle implicazioni potenzialmente più ampie). Per la cronaca io facevo parte di un gruppo di persone che stavano tentando di spingere l'azienda a rendere pubblica almeno una parte del suo codice.

Sentivo la pressione del fatto di dover mantenere nella comunità *open source* la mia posizione di persona affidabile da un punto di vista sia tecnologico sia etico. Per me era importante non schierarmi con l'una o con l'altra azienda Linux in concorrenza. No, non mi stavo vendendo quando accettai le stock option che la Red Hat fu tanto gentile da offrirmi a titolo di ringraziamento. Mi sembrò invece giusto rifiutare l'offerta dell'imprenditore di Londra che mi offriva 10 miliardi di dollari solo per prestare il mio nome come membro del consiglio d'amministrazione della sua nascente società Linux. Non poteva credere che io rifiutassi tutti quei soldi. Una cosa del tipo: «Qual è la parte che le sfugge nell'espressione dieci miliardi di dollari?»

Non avevo pensato di dover affrontare problemi del genere. E la nuova popolarità di Linux portò con sé dei grattacapi non solo per me ma anche per tutta la comunità virtuale. In effetti, quando nel 1998 il codice *open source* catturò l'attenzione di tutto il mondo, uno dei grandi dibattiti verteva sul suo stesso nome. Fino ad allora ci eravamo riferiti al fenomeno della condivisione di software e all'utilizzo di licenze come la GPL, come «free software:», e in generale ci riferivamo al «movimento del free software». Il termine affondava le proprie radici nella Free Software Foundation, fondata da Richard Stallman nel 1985 per promuovere progetti free software come GNU, la versione free di Unix lanciata da Stallman stesso. All'improvviso divulgatori come Eric Raymond scoprirono che i giornalisti non avevano le idee chiare. La parola «free»[9] significava che non costava nulla? Che non aveva costrizioni? Aveva a che fare con il concetto astratto di libertà? Si scoprì che Brian Behlendorf, che parlava ai giornalisti a nome dell'Apache, si trovava a che fare con frustrazioni di questo tipo.

[9] Ricordiamo che in inglese il termine free significa al contempo libero e gratuito. [n.d.t]

Dopo settimane di scambi di e-mail private (di cui io non fui un partecipante attivo ma semplicemente un osservatore, dato che non mi interessavano gli aspetti politici), venne raggiunto un accordo. Avremmo usato la parola «open» (aperto) invece del termine «free». Fu così che il movimento del free software divenne il movimento *dell'open source* (almeno per chi preferiva vederlo come un movimento, cosa a mio parere non lontana dalla realtà). In ogni caso la Free Software Foundation mantiene ancora oggi il proprio nome e Richard Stallman ne è il cervello pensante.

Essendo uno dei leader *de facto* del movimento, ero sempre più ricercato. Ogni volta che suonava il mio telefono alla Transmeta - e in quei giorni suonava continuamente - poteva voler dire due cose: o un giornalista voleva intervistarmi o gli organizzatori di una conferenza mi volevano come oratore. Per diffondere la conoscenza *dell'open source* e di Linux, mi sentivo obbligato a fare entrambe le cose. Prendete un timido mago della matematica, mettetelo nel circuito saluta&sorridi per una causa populista, e ne avrete fatto un eroe popolare. Dimenticate quello che ha detto Eric Raymond sul fatto che io ero evidentemente meno strano di molti hacker. Una buona parte del mio fascino (o comunque vogliale chiamarlo) stava nel fatto che io *non* ero Bill Gates.

I giornalisti sembravano adorare il fatto che, mentre Gates viveva in una supervilla high-tech in riva al lago, io inciampavo nei giocattoli delle mie bambine nella nostra nuova casa, una *ranch house* con tre camere da letto in una villetta bifamiliare con un pessimo impianto idraulico nella noiosa Santa Clara. E per di più me ne andavo in giro con una Pontiac qualsiasi. E rispondevo personalmente al telefono. Come avrebbero potuto non adorarmi?

A mano a mano che Linux iniziava a essere considerato una vera minaccia per Microsoft e siccome Microsoft aveva i ben noti problemi con l'antitrust una vera minaccia le faceva sicuramente comodo - la stampa balzava su ogni nuovo sviluppo come se fosse stata la seconda guerra mondiale. Qualcuno lasciò trapelare il «Documento di Halloween», un memo interno di Microsoft in cui si diceva che l'azienda era preoccupata per Linux. Ben presto i giornali riportarono una dichiarazione di Steve Balmer: «Certo che sono preoccupato. Il fatto era che per quanto la Microsoft traesse dei benefici dall'enfatizzare la concorrenza che Linux faceva a Windows NT - questa stava continuando a crescere.

Io non avevo bisogno di mettermi su un piedestallo e dire cose orribili sulla Microsoft. Perché avrei dovuto farlo? Le cose andavano avanti da sole, e giocavano a favore di Linux. I giornalisti adoravano tutta quella faccenda. Il Benvoluto David contro il Golia Monopolistico e Maligno. E, dato che sto dicendo tutta la verità e nient'altro che la verità, in effetti mi piaceva parlarne con i giornalisti. Mi piace parlare male dei giornalisti, ma la maggior parte delle mie interviste sono state divertenti. I giornalisti di solito erano interessati alla nostra storia: a chi non piace schierarsi dalla parte del più debole?

Una volta assunta la propria dose della storia «l'Ameba-che-aveva-distrutto-Microsoft» (nota: nell'interesse della verità dichiariamo qui che questa frase è stata controllata ortograficamente con un prodotto Microsoft), i giornalisti volevano capire il concetto di *open source*. Il nostro messaggio faceva sempre meno fatica a passare, perché la gente ora poteva vederne un esempio in azione. Ciò che poi sembrava

affascinarli di più era l'amministrazione di Linux. Non riuscivano a capire come il più grande progetto collaborativo nella storia dell'umanità potesse essere gestito con tanta efficienza quando un'azienda con trenta dipendenti tende normalmente a trasformarsi in una specie di incasinatissimo pollaio.

Qualcuno coniò l'espressione «Dittatore Benevolo per descrivere il modo in cui gestivo la cosa. La prima volta che sentii quest'espressione riuscii a pensare solo a un generale dai baffoni scuri di una qualche nazione soleggiata, intento a distribuire banane a una folla affamata. Non so se mi si addica l'immagine del Dittatore Benevolo. Io controllo il kernel di Linux, ovvero le sue fondamenta, perché sino a ora tutti coloro che hanno a che fare con Linux si sono fidati di me più che di chiunque altro. Il mio metodo per gestire questo progetto con centinaia di migliaia di sviluppatori è lo stesso di quando digitavo il mio codice nella mia camera da letto: anziché delegare in modi proattivi, aspetto che le persone si facciano avanti e si offrano volontarie. Ho iniziato quando mi sono liberato delle responsabilità che trovavo meno interessanti, come il codice a livello utente. Altre persone si sono fatte avanti e si sono offerte di occuparsi dei sottosistemi. Tutto quanto filtra fino a me attraverso la gestione di questi sottosistemi.

Io posso approvare o disapprovare il loro lavoro, ma perlopiù lascio che le cose accadano in modo naturale. Se due persone gestiscono delle cose simili, li accetto entrambi per vedere quale viene utilizzato. A volte vengono utilizzati entrambi, ma finiscono per prendere strade diverse. Una volta c'era una forte concorrenza tra due persone che insistevano a mandarmi patch contrapposte. Io rifiutai di accettare le patch di entrambi, finché uno dei due sviluppatori non perse interesse per la cosa. Re Salomone avrebbe gestito le cose allo stesso modo, se si fosse trovato a gestire un asilo.

Dittatore benevolo? No, sono solo pigro. Cerco di gestire senza prendere decisioni e lasciando che le cose accadano in modo naturale. E così che si ottengono i risultati migliori.

Il mio approccio finì sulle prime pagine dei giornali.

Ma l'ironia di tutta la faccenda è che mentre la mia gestione di Linux conquistava l'attenzione della stampa, quando per un breve periodo diventai dirigente alla Transmeta fui un assoluto disastro. A un certo punto venne deciso che io avrei dovuto gestire un team di sviluppatori. Feci fiasco. Come sa chiunque si sia avventurato dentro quella specie di discarica che è il mio ufficio, io sono totalmente disorganizzato. Avevo dei problemi a gestire le riunioni settimanali sui progressi del progetto, la valutazione delle prestazioni, le linee di azione. Dopo tre mesi fu evidente che il mio stile dirigenziale non era in alcun modo utile alla Transmeta, nonostante le lodi che i giornalisti mi tributavano per come gestivo Linux.

Nel frattempo la stampa continuava a battere su un altro argomento: la frammentazione. Chi ha seguito la triste e difficile storia di Unix sa delle interminabili lotte tra i suoi rivenditori. La questione sorse nel 1998: la storia si sarebbe ripetuta anche nel mondo di Linux? La mia risposta era sempre che per quanto ci fossero indubbiamente delle lotte tra i rivenditori di Linux, queste non potevano portare alla frammentazione che aveva sempre impedito il successo finale di Unix. Il problema di Unix è che i rivenditori concorrenti avevano sprecato anni a

implementare funzioni simili, solo perché non avevano accesso alla stessa base di sorgenti. Lo sviluppo indipendente delle stesse funzioni non è solo costato a Unix degli anni di lavoro, ma lo ha anche portato a sanguinose lotte interne. Certo, dicevo alla stampa, i rivenditori di Linux non si ritrovano tutti quanti a festeggiare insieme il Natale, ma c'è e continuerà a esserci meno frammentazione nella comunità Linux che in quella Unix, perché anche i rivenditori Linux che non hanno buoni rapporti possono accedere alla stessa base di sorgenti e riutilizzare il lavoro degli altri. Il codice sorgente è un magazzino a cui tutti possono attingere.

Più i giornalisti iniziavano ad afferrare questi concetti, più mi piaceva incontrarli. (A differenza dei giornalisti che ricordavo dai tempi della mia gioventù a Helsinki, la maggior parte di quelli che incontravo negli Stati Uniti negli anni Novanta non erano ubriachi.) Mi faceva piacere in particolare avere la possibilità di discutere con loro.

Tenere discorsi in pubblico era tutta un'altra faccenda. Non sono esattamente un istrione. Ricordate? Io sono quello che per tutta l'adolescenza non è praticamente uscito dalla propria camera da letto. Non sono mai stato molto bravo nemmeno a *scriverli,* i discorsi. Così aspettavo sempre la notte prima della conferenza per prepararli.

In qualche modo questo sembrava non essere un problema. Di solito salivo sul palco e la gente si alzava in piedi ad applaudirmi prima ancora che avessi aperto bocca. Non vorrei sembrare un ingrato, ma l'ho sempre trovata una situazione imbarazzante. Qualsiasi cosa tu possa dire a quel punto sembra sbagliata, anche il mio solito «Grazie, ma adesso sedetevi, per favore.» Sono disposto ad accettare suggerimenti al riguardo.

Ma non tutte le telefonate arrivavano da giornalisti o organizzatori di conferenze. Una sera me ne stavo seduto in casa con Tove. Stavamo leggendo qualcosa alle bambine. Squillò il telefono.

Risposi: «Torvalds».

«Uh. Sei il tizio di Linux?»

«Sì.»

Due secondi di silenzio. *Click.*

Un'altra sera un tizio di Las Vegas mi telefonò a casa per convincermi a entrare in un affare che aveva a che fare con delle T-shirt di Linux.

La soluzione più ovvia sarebbe stata quella di farmi assegnare un numero di telefono privato. Quando ci eravamo trasferiti in California non ci avevo nemmeno pensato, perché costava molto di più che averne uno pubblico. Poi capii che, come dicono le mamme, a volte chi più spende meno spende. E adesso ho un numero privato. Una volta, prima di tutto ciò, David perse il mio numero di casa e chiamò il servizio informazioni della compagnia telefonica. Chiese il mio numero e l'operatore che glielo diede disse sbalordito: «Ha un numero pubblico? Con tutti i suoi milioni?»

E invece no. Niente milioni. Milioni di utenti Linux, questo sì. Ma niente milioni di dollari per Linus.

E la cosa non era affatto un problema.

VII.

Spesso mi sveglio pensando che sono il bastardo più fortunato del mondo. Non ricordo se mercoledì 11 agosto 1999 fu una di queste giornate, ma avrebbe dovuto esserlo.

Era il secondo giorno della fiera Linux World che si svolgeva al San Jose Convention Center. Dirk Hohandel, che era l'amministratore delegato di SuSE ed era arrivato dalla Germania apposta per quella fiera, aveva passato la notte sul divano letto del nostro salotto. Lo conoscevo da anni. È uno dei tizi del vecchio xfree86 e si occupa degli aspetti grafici di Linux. Ed è anche il padrino di Daniela. Mi svegliai, preparai il cappuccino per Tove e Dirk, lessi il «San Jose Mercury News» dalla prima all'ultima pagina saltando solo lo sport e gli annunci - come faccio sempre - e poi ci ammassammo nella Toyota Rav4 per fare le dieci miglia fino al centro di San Jose.

Ricordo di avere stretto un sacco di mani.

Era il giorno in cui Red Hat si sarebbe quotata. Qualche anno prima l'azienda mi aveva dato delle stock option e di recente mi avevano spedito dei documenti che io non avevo neppure letto. Erano da qualche parte, in una pila di carte accanto al mio computer. Ricordo che desideravo molto che la Red Hat andasse bene. Non era tanto che fossi in agitazione per le mie stock option: in realtà non sapevo nemmeno bene cosa volesse dire. Ero eccitatissimo per un'altra ragione. Sotto molti punti di vista, il successo dell'IPO sarebbe stato una conferma per Linux. Quindi quella mattina ero un po'nervoso, ma non ero il solo. Il mercato andava male da settimane e la gente si chiedeva se Red Hat sarebbe uscita viva dalla IPO.

Alla fine accadde quello che si chiamava «evento liquità». Ci giunse voce che l'offerta pubblica iniziale di Red Hat era partita a 15 dollari. O forse 18? Non ricordo. L'importante era che alla fine della prima giornata di scambi le azioni erano arrivate a 35 dollari. Non è che fosse una performance da record, ma andava bene così.

Ricordo di essere tornato a casa in auto con Tove e Dirk. All'inizio mi sentii sollevato. Poi pensai ai soldi e mi sentii sempre più eccitato. Solo quando restammo bloccati nel traffico diretto verso nord della Highway 101 mi venne in mente che in un solo giorno ero passato da praticamente zero a mezzo milione di dollari. Il mio cuore iniziò a battere come un tamburo. Provavo una specie di euforia mista a incredulità.

Non sapevo niente di azioni e decisi che dovevo capire cosa avrei dovuto fare. Cosi telefonai a Larry Augustin, l'amministratore delegato di VA Linux. Gli dissi che era l'unica persona che conoscevo che sapesse qualcosa di azioni. Le mie esatte parole furono: «Tu hai un tuo brocker o qualcuno di cui ti fidi? Perché non vorrei venderle su e-Bay.»

La Red Hat mi aveva dato delle option, non esattamente delle azioni. Non sapevo cosa fare. Sapevo che potevano essere vincolate per un certo periodo, ma non sapevo se la cosa valeva anche per me e non avevo pensato alle conseguenze fiscali. Larry, che è bravo in queste cose e conosce un sacco di gente, mi mise in contatto con un tizio di Lehman Brothers che non avrebbe dovuto lavorare per me perché non ero un

cliente abbastanza grosso. Mi promise che avrebbe scoperto cosa avrei dovuto fare. Nel frattempo, due giorni dopo la IPO, ricevetti una e-mail da qualcuno dell'ufficio Risorse Umane della Red Hat, o dal loro avvocato, in cui mi diceva qualcosa su uno split delle azioni avvenuto prima dell'IPO. Non avevo idea di cosa significasse. Così presi la busta con tutta la documentazione sulle stock option che non mi ero mai curato di leggere e trovai tutto scritto nero su bianco: le mie opzioni erano magicamente raddoppiate.

Il mio mezzo milione di dollari era in realtà *un milione* di dollari!

La stampa ha dato di me l'immagine di un geek-per-le-masse altruista e votato alla povertà. In realtà in quel momento stavo impazzendo di gioia.

Ecco, l'ho detto.

Mi sedetti e lessi tutte le carte della Red Hat. Sì, ero soggetto a un vincolo di 180 giorni.

Avete un'idea di quanto possano essere lunghi 180 giorni quando siete per la prima volta dei potenziali miliardari?

Ora avevo un nuovo sport (o forse è meglio dire uno sport, punto e basta): seguire il prezzo delle azioni Red Hat, che continuarono a salire per i sei mesi che seguirono. Salivano gradualmente, qualche volta fecero dei balzi. Salivano. Salivano. A un certo punto la Red Hat fece addirittura un altro split. All'apice delle quotazioni le mie opzioni valevano 5 milioni di dollari.

La Red Hat era partita relativamente bassa ed era salita inarrestabilmente a mano a mano che Wall Street - nel pieno di una storia d'amore con qualsiasi cosa fosse anche solo lontanamente collegata a internet - «scopriva» Linux. Nei freddi mesi che chiusero il 1996 eravamo la Notizia del Momento. I guru della finanza alla televisione e sulla stampa non ne avevano mai abbastanza di questo piccolo e folle sistema operativo che prometteva di mandare Microsoft con le gambe all'aria. Il mio telefono squillava continuamente. Il massimo del parossismo venne raggiunto con l'IPO di VA Linux, il 9 dicembre. Fu un evento che andò oltre ogni aspettativa.

Io e Larry Augustin andammo a San Francisco per stare negli uffici della First Boston Credit Suisse durante la IPO. Io indossavo i miei soliti vestiti: una maglietta promozionale e un paio di sandali. Portammo le mogli e i bambini. Era una scena da matti, con quei poppanti che correvano tra i piedi di quegli yuppie tutti in tiro.

Accadde tutto all'improvviso. Le cifre schizzavano sui monitor: in quella prima giornata di scambi le azioni di VA Linux si vendevano intorno ai 300 dollari. Non si era mai sentita una cosa del genere. Anche se non avessimo visto le cifre, avremmo probabilmente capito che era un record da come gli yuppie erano ipnotizzati dalla CNN e dal canale finanziario Bloomberg. Da parte sua Larry si mantenne calmo, come sempre. Non credo che abbia battuto ciglio per tutto il tempo. Per quanto mi riguarda non saprei dire, ero troppo impegnato a inseguire le bambine.

Anche gli abitanti delle foreste del Madagascar hanno probabilmente avuto notizia di quanto Larry sia diventato ricco dopo tutto questo. Era arrivato a San Francisco praticamente senza un soldo e se ne tornava nella Silicon Valley con in tasca qualcosa come 1 miliardo e 600 milioni di dollari. E, come la stampa ha continuato a sottolineare, non aveva nemmeno trentanni.

Io avevo ricevuto della azioni e delle option di VA Linux.

Come per la Red Hat, non potevo venderle per sei mesi. Ma a differenza della Red Hat, che aveva avuto una crescita costante, le VA Linux non potevano che scendere. Dopo l’apertura da record, le azioni continuarono a calare per un anno, raggiungendo il punto più basso a 6,62 dollari. Furono in parte vittime dell’inversione di tendenza del mercato che ad aprile colpì la maggior parte delle azioni tecnologiche. Ma c’era anche il fatto che la moda mediatica di Linux era finita con l’arrivo della primavera. Dato il vincolo delle mie VA Linux, non potevo avvantaggiarmi del periodo di vacche grasse. Fu psicologicamente molto più duro seguire l’andamento dell’azienda di quanto fosse stato con la Red Hat, sapere che ogni mattina mi sarei sveglialo un po’meno ricco della sera prima.

Ma continuavo a sentirmi il bastardo più fortunato del mondo.

Una sera di gennaio Linus arriva in auto al mio ufficio di Sausalito. Dopo avermi preso in giro per il fatto che uso un sistema operativo Macintosh e non Linux, si siede a leggere la prima bozza di una lunga prefazione che ho scritto per il libro. È in prima persona, dal suo punto di vista. Io gli sto a non più di dieci centimetri. La sola volta che Linus dice qualcosa mentre legge è quando arriva a una frase in cui si dice che non si sarebbe mai aspettato di diventare l'unica super star globale prodotta dalla Finlandia oltre a Jean Sibelius e la renna Nikki. Dopo una decina di minuti finisce la prefazione e il suo unico commento è: «Ragazzi, scrivi frasi lunghe, eh?» Passiamo un paio d'ore ad accorciare le frasi e a cambiare in parte il linguaggio, introducendo le parole che lui usa di solito. Stiamo imparando a collaborare anche sul lavoro, visto che abbiamo già ampiamente dimostrato di saperlo fare quando c'è da perdere tempo. Alla fine quella prefazione l'abbiamo buttala via.

Poi Linus prova (senza riuscirvi) a migliorare la risoluzione del mio schermo ultrapiatto. L'anno scorso era lo schermo più all'avanguardia e io lo vedevo come una specie di status symbol. «Come fai a leggere su quella roba?», mi chiede. Non riesce a migliorare la risoluzione in modo che lo soddisfi. Poi prende un pezzo di carta e inizia a disegnare dei diagrammi e a spiegarmi come funziona un monitor. A un certo punto gli dico: «Ehi, facciamoci un sushi.»

«Questa faccenda dei soldi mi sta facendo impazzire», dice Linus. «Il fatto di aspettare che finisca il periodo di vincolo delle azioni, dico. È come avere un sacco di soldi ma non avere un sacco di soldi. Non faccio che pensarci.»

Io ordino del sake. Lui del succo di frutta, perché deve guidare.

«Fino a oggi non abbiamo mai avuto più di 5000 dollari sul nostro conto. A parte delle azioni e dell'altra roba che abbiamo messo via come risparmio e che non possiamo toccare, erano tutti lì i soldi che avevamo da spendere. Così adesso sulla carta ho tutti questi soldi e...»

«Di quanti soldi stiamo parlando? Un paio di milioni di dollari?»

«Direi una ventina di milioni. È il valore delle azioni della IPO di VA Linux, se non crollano. Ma non posso avere i soldi finché non finisce il vincolo di sei mesi. Anzi, adesso sono cinque mesi.»

«Non vedo dove sta il problema. Devi aspettare cinque mesi prima di comprarti una bella casa? Non vorrei sembrarti poco comprensivo, ma...»

«Eh, lo so, all'inizio sembrano abbastanza soldi per poterci comprare qualunque casa vogliamo. Ma abbiamo bisogno di una casa con cinque camere da letto e vogliamo abbastanza terreno attorno alla casa così possiamo sentire gli animali e poi al lavoro gioco a biliardo tutti i giorni così avremo bisogno di una stanza abbastanza grande per metterci un tavolo da biliardo. E vogliamo una dependance per quando vengono a trovarci i genitori di Love, oppure per far venire gli amici di mia sorella dalla Finlandia e farli stare da noi qualche mese per aiutarci con le bambine. È buffo, abbiamo avuto Patricia quando ci siamo trasferiti dalla Finlandia agli Stati Uniti. Daniela l'abbiamo avuta quando ci siamo trasferiti

dall’appartamento alla villetta, e...»

«Quindi state cercando di avere un altro bambino?»

«Be’, lasciamo che la natura faccia il suo corso.»

«A casa mia questo si traduce con ”Stiamo cercando di avere un bambino”, amico.»

«Ok. Così avremo bisogno di più spazio. Siamo andati in un paio di agenzie e le case che avevano erano incredibilmente care. Voglio dire, ti ritrovi con 20 milioni di dollari e pensi una cosa tipo Wow, mi posso permettere tutte le case del mondo. Invece siamo andati a vedere una casa che costava 1 milione e duecentomila dollari a Woodside, ma non aveva terreno attorno ed era ridotta da far schifo. La casa più bella che abbiamo visto costava 5 milioni. Ma se hai 20 milioni di dollari, devi calcolare che la metà se ne andrà in tasse. Così hai 10 milioni di dollari su cui fare conto, ma le tasse su una casa del genere saranno 60.000 dollari all’anno, quindi devi mettere da parte i soldi per pagarle. Non lo so. Sarà l’unica volta nella mia vita in cui mi ritroverò così tanti soldi in tasca e non voglio fare il passo troppo lungo e poi non potermi permettere di vivere in casa mia. E non vogliamo nemmeno avere un mutuo che ci incomba sulla testa.»

«Non è che tu mi faccia proprio pena, sai. Innanzitutto se la Transmeta avrà una buona IPO tu ci guadagnerai bene.»

«Sì. Ma sono solo uno junior engineer. Non mi daranno moltissime azioni. E non ho uno stipendio altissimo. »

«Linus, potresti andare da qualsiasi investitore in questo paese e ottenere qualsiasi cifra...»

«Credo tu non abbia torto.»

VIII.

Questa è la parte del libro in cui vi comunico le mie regole d'oro. La Numero Uno dice: «Fai agli altri quello che vorresti fosse fatto a te.» Se segui questa regola, saprai sempre come comportarti in ogni situazione. La Numero Due dice: «Sii fiero di ciò che fai.» La Numero Tre: «E divertiti a farlo».

Naturalmente non è sempre facile essere fieri e divertirsi. Un mese prima della IPO di VA Linux non mi riuscì nessuna delle due cose quando tenni un discorso al Comdex del 1999, a Las Vegas. Il Comdex, come tutti sanno, è la più grande e maledetta fiera che l'umanità abbia mai visto. Per la maggior parte di quella settimana la sonnolenta città di Las Vegas, Nevada, si trasforma in una calamita per qualsiasi prodotto high-tech concepibile, nonché per la massa di persone che lo vorrebbero vendere o comprare. É l'unico momento dell'anno in cui a Las Vegas puoi abbassare il finestrino del tuo taxi, sporgerti, chiedere a una puttana a caso: «A che ora è il discorso d'apertura?» e aspettarti di ricevere la risposta esatta.

Era importante che gli organizzatori della fiera avessero chiesto al benevolo dittatore del Pianeta Linux di tenere un discorso al Comdex. Era il modo del settore informatico di riconoscere che Linux era una forza che andava riconosciuta.

Bill Gates aveva tenuto un discorso la domenica, la prima sera della fiera. Aveva attratto un pubblico numerosissimo nella sala da ballo del Venetian Hotel, che ha più o meno le dimensioni di un IKEA. Il pubblico - gente che voleva sapere cosa avesse da dire sulla faccenda dell'antitrust, di grande attualità in quei giorni, o che voleva semplicemente poter dire ai propri nipoti di avere visto in carne e ossa l'uomo più ricco del mondo - aveva fatto la fila per ore nel seminterrato dell'enorme centro conferenze dell'hotel. Il discorso di Gates era iniziato con una barzelletta sugli avvocati, per passare poi a dimostrazioni ben coreografate della tecnologia web di Microsoft e a elaborati interventi video, in uno dei quali Gates imitava Austin Powers (con vestito anni Sessanta e tutto quanto). Il pubblico si era piegato in due dalle risate.

Io non c'ero. Stavo accompagnando Tove a comprare un costume da bagno.

Ma la sera dopo tenni il mio discorso nella stessa sala. Avrei preferito andare a fare shopping. Be', non è *proprio* vero...

Non è che fossi impreparato. Di solito scrivo i miei discorsi il giorno prima, ma questa volta avevo decisamente giocato d'anticipo. L'intervento era il lunedì sera e il sabato avevo scritto tutto quanto e avevo preparato il computer per la proiezione delle slide. Era tutto a posto. Avevo addirittura messo il mio discorso su tre dischetti diversi, tanto per andare sul sicuro nel caso uno fosse stato difettoso. L'unica cosa che odio più di parlare in pubblico è parlare in pubblico quando qualcosa va male. Avevo messo il mio discorso su internet, nel caso che *tutti* i dischetti fossero difettosi.

Sullo Strip c'era un ingorgo stradale (naturalmente causato dal Comdex), così arrivammo al Venetian Hotel solo mezz'ora prima del mio intervento. Ero con Tove, le bambine e dell'altra gente della fiera. Quando alla fine entrammo nell'edificio,

avemmo dei problemi a entrare nella zona dietro il palco perché uno degli organizzatori aveva fatto casino con i badge della sicurezza. Insomma: tutto stava andando per il peggio.

Alla fine entrammo. Sarei stato nervoso se avessi dovuto parlare davanti a quaranta persone, figurarsi davanti al pubblico più numeroso di tutta la mia vita. Poi accadde.

Scoprii che il computer, che due giorni prima era stato preparato con tanta cura, era scomparso. Era una cosa folle. Qualcuno accennò al fatto che la gente aveva iniziato a mettersi in coda al piano di sotto quattro ore prima dell'orario d'inizio del discorso e che la zona d'attesa era stracolma. Nel frattempo noi correvamo in giro come delle trottole, alla ricerca di quella macchina.

Era un normalissimo sistema desktop con installato Star Office, uno dei pacchetti office di Linux. Io avrei dovuto semplicemente infilarvi i miei dischetti e via. Era stato tutto preparato in modo che non ci dovessero essere cavi da attaccare. Ma il computer era scomparso. Pare che vi fosse stata attaccata sopra un'etichetta sbagliata o qualcosa del genere, per cui era stato rispedito indietro. Fortunatamente avevo con me il mio portatile e su quel computer avevo sia il file del mio discorso sia Star Office.

Dato che si trattava del mio portatile personale, non avevo tutti i font giusti. Il che voleva dire che in tutte le mie slide mancava l'ultima riga. Quando me ne resi conto pensai: Chi se ne frega? Mica mi ammazzeranno per una cosa del genere, no? Poi dovemmo attaccare tutti i cavi. Nel frattempo iniziarono a far entrare la gente, prima ancora che fosse tutto installato. Voglio dire: io ero lì che provavo ancora a far funzionare quel coso mentre una fiumana di gente irrompeva in quella sala sconfinata, occupandone ogni posto libero e poi piazzandosi in piedi lungo le pareti. Fortunatamente mi riservarono una standing ovation ancora prima che aprissi bocca.

Iniziai con un riferimento alla barzelletta sugli avvocati che Bill Gates aveva usato per aprire il proprio discorso. Dedicai una frase all'attività della Transmeta, all'epoca ancora sotto segreto. La stampa aveva fatto girare la voce che avrei approfittato del discorso al Comdex per annunciare (finalmente) il chip della Transmeta. Ma non eravamo ancora pronti. La parte principale del mio discorso consisteva semplicemente in un'esposizione dei benefici dell'open source. Non ero dell'umore di fare battute, come faccio di solito. A un certo punto Daniela - seduta con Tove e Patricia in prima fila - scoppiò a piangere tanto forte che probabilmente la sentirono in tutti i casinò e i locali di striptease di Las Vegas.

Quel discorso non finirà probabilmente nella storia dell'oratoria. Più tardi qualcuno cercò di tirarmi su informandomi che anche Bill Gates la sera prima era visibilmente nervoso. Ma la sua organizzazione aveva funzionato come un orologio svizzero. Il problema era che aveva il Dipartimento della Giustizia che gli alitava sul collo. Mi sa che ne sono uscito meglio io.

Sembrava una buona strategia da Manuale del Giornalismo: trova la persona che sta aspettando di più per ascoltare il discorso di Linus e mettiti in coda vicino a lui (e sul fatto che si tratti di un lui non c'è alcun dubbio). Quale modo migliore per raccogliere informazioni sull'orda di nerd che seguono Linus come se fosse una specie di Dio ipersponsorizzato?

Alle cinque del pomeriggio sono sulla scala mobile che porta alla «Woodstock dei Geek». All'estremità della lunghissima coda serpeggiante c'è uno studente di informatica del Walla Walla College che accetta ben volentieri che mi unisca a lui. Finora ha aspettato due ore e mezza per vedere Linus, e aspetterà altre due ore e mezza prima di entrare nell'auditorium. I suoi compagni di corso, che si trovano in coda subito dietro, sono arrivati una mezz'ora dopo di lui. Sono arrivati dallo stato di Washington con uno dei loro professori e dormono nella palestra di un liceo del posto. Sembra che abbiano avviato tutti quanti una propria attività di web design. Hanno diviso l'universo degli adulti in due gruppi - gli hacker e i doppiopetto -e continuano a indicare dei membri della seconda categoria che stanno in coda dicendo cose tipo «Ragazzi, guardate quei doppiopetto laggiù» come i ragazzi di una confraternita tipo Delta Tau Chi potrebbero indicare la spiaggia durante le vacanze di Pasqua e dire «Ragazzi, guardate quelle gnocchette laggiù». É proprio come i ragazzi del Delta Tau Chi continuano a darsi il cinque e scambiarsi insulti, anche se gli insulti nel loro caso riguardano schede madri e gigabyte.

E poi parlano di Linus. Il suo nome viene sempre pronunciato in lettere maiuscole, come nella frase «LINUS non lavorerebbe per una società che non fosse disposta a passare all'open source. Non lo farebbe proprio. » Conoscono a menadito siti come Slashdot o altri in cui le voci sulle attività della Transmeta giravano come fossero stati i particolari più piccanti della vita sentimentale di una stellina di Hollywood. Questa mania, queste speculazioni, questo fascino non si danno solo tra i fan più scatenati, quelli arrivati per primi.

Faccio un salto alla toilette degli uomini e prendo posto all'unico orinatoio libero, interrompendo una conversazione in corso. «Questo discorso sarà bello noioso in confronto a quello di Gates», dice il tizio alla mia sinistra.

«Cosa ti aspetti?», risponde l'altro. «Linus è un hacker, non un doppiopetto. Voglio dire, dagli tregua.»

Quando alla fine entriamo nell'auditorium per un qualche motivo non ci ritroviamo nelle prime file ma verso il fondo della parte centrale. Il mio compagno di coda del Walla Walla dimentica per un istante l'eccitazione di vedere dal vivo il proprio eroe e dà via di testa all'idea di non essere sotto il palco, dove dovrebbe stare. Inizia a indicare i doppiopetto in sala. Anche se siamo ad almeno cinquanta metri dalla prima fila riusciamo a vedere Linus sul palco ancora buio, seduto davanti a un computer. Digita velocemente sulla tastiera, circondato da alcune persone dell'organizzazione. Cosa starà succedendo lassù? Qualche dimostrazione software dell'ultimo momento?

Alla fine Linus e gli altri lasciano il palco. Viene presentato il direttore esecutivo di Linux International Maddog (Jon Hall). Il mio nuovo amico del Walla Walla è evidentemente emozionato. «Guarda che barba», dice. Poi Maddog dice di essere fiero di presentare un uomo che per lui è come un figlio. Linus riemerge sulla scena e Maddog lo abbraccia come un orso. Anche dalla posizione da quattro soldi in cui mi trovo riesco a capire quanto Linus sia emozionato.

«Volevo iniziare con una barzelletta sugli avvocati, ma mi hanno battuto sul tempo», dice riferendosi al fortunato intervento del giorno precedente, tenuto da un Bill Gates assediato dai doppiopetto dell'antitrust: «Conoscete una bella barzelletta sugli avvocati?»

Poi dedica una sola frase alle operazioni segrete della Transmeta. Il resto del suo discorso consiste nella lettura a cento all'ora delle frasi che appaiono sulle slide proiettate sopra di lui, dichiarazioni sulla crescente importanza dell'open source. Niente, di sorprendente. Niente di nuovo.

Parla con un tono piatto, stanco ma amichevole. A un certo punto una delle sue bimbe scoppia a piangere.

Nel bel mezzo di una frase Linus dice: «Quella è mia figlia». Guardando i monitor si vedevano le luci di scena riflettere i rivoli di sudore che gli solcavano la fronte.

Alla fine il pubblico fa la fila per porgli qualche domanda. Si rifiuta di dire quale sia il suo word processor Linux preferito. Quando qualcuno gli chiede quanti pinguini di peluche abbia a casa, risponde: «Un bel po'». Un membro del pubblico chiede se gli piaccia vivere in California. Linus risponde con una tirata sul clima: «È novembre e porto ancora i pantaloni corti. A Helsinki a quest'ora avrei già perso i miei gioielli della corona.» Un fan si impadronisce del microfono per le domande del pubblico e annuncia semplicemente: «Linus, sei il mio eroe». Al che Linus, come se avesse sentito quella frase un milione di volte e un milione di volte avesse risposto, dice solo: «Grazie».

Dopo che le domande sono finite, centinaia di persone accorrono nella zona sotto il palco, dove Linus stringe tutte le mani possibili.

IX. La rivoluzione di Linux è finita?
di Scott Berinato, «PC Week»

«Grazie per avere chiamato. La Rivoluzione è finita. Se desiderate ulteriori informazioni su Linux, siete pregati di digitare 1...»

Pare che Linus Torvalds abbia una segreteria telefonica, il che deve voler dire che l'affare Linux è diventato mainstream. Quindi scordatevi la rivoluzione e tornate al lavoro sul vostro desktop Windows.

C'era un tempo in cui i giornalisti potevano chiamare l'inventore del sistema operativo Linux nel suo ufficio presso la (teoricamente) avventurosa Transmeta Corp, digitare il suo interno e ricevere un familiare « Torvalds» dalla persona che stava dall'altra parte del filo. Era paziente e rispondeva alle vostre domande. Quando non aveva tempo ve lo diceva. A volte vi diceva che stavate facendo domande insensate, da neofita della programmazione. Ma rispondeva al telefono. Oggi, quando chiamate la Transmeta e digitate il suo interno, vi accoglie una gradevole voce femminile. «Grazie per avere chiamato Linus Torvalds. Questa casella vocale non accetta messaggi. Per contattarlo siete pregati di inviare un fax al numero...»

Cosa? E a quel punto iniziate a pensare: non mi risponderà. Ne ha avuto abbastanza. È una celebrità e ottenere una breve intervista con lui ora sarà come ottenere una breve intervista con l'altra, grande celebrità dell'informatica. La signorina al telefono spara un numero di fax e voi state già pensando di digitare la classica combinazione 0-# per parlare con una centralinista...

«Il nostro centralino non prende messaggi per lui e non dispone dell'agenda dei suoi impegni.» Oh, oh... Che carina... Il peggio. «Ma saremo lieti di fargli avere il vostro fax.» Uh, uh... E Bill sarà lieto di fare a pezzettini la Microsoft per far piacere a David Boies.

Ok, la rivoluzione di Linux forse non è finita, ma come tutte le rivoluzioni il casino creativo degli inizi è stato sostituito da un'algida gentilezza aziendale. La new wave suburbana ha soppiantato il punk urbano. Il ricco latifondista delle colonie americane è figlio del pover'uomo ipertassato dalla regina d'Inghilterra. (Peraltro in seguito i ricchi latifondisti cercarono di imporre agli uomini di frontiera una tassa sul whisky non molto diversa da quella sul tè che qualche anno prima era stato imposta a loro stessi.)

In effetti era probabilmente ora che Linus facesse un passo indietro. Era inevitabile, dato il numero di telefonate di giornalisti e la folle quantità di argomenti su cui veniva interpellato.

Prendete la conferenza stampa alla Linux World Expo di San Jose, all'inizio di questo mese. Torvalds - che aveva accettato di tenere questa conferenza semplicemente perché non aveva tempo per rispondere a tutte le singole richieste di intervista - all'inizio ha dovuto sparare le solite risposte alle solite

domande. L'open source può funzionare in ambito aziendale? Sta cercando di dominare il software come fa Bill Gates? Cosa pensa della Microsoft? Cos'è l'open source? Cos'è Linux? Perché un pinguino?

Torvalds a quel punto stava ormai chiaramente attingendo a una serie di risposte preconfezionate. Pensate a Tim Robbins in Bull Durham: «Devo solo uscire là fuori e dare il 110 per cento per cercare di aiutare la squadra...»

E oltre a essere ridondanti, le domande dei giornalisti che non appartenevano al mondo della tecnologia svariavano paurosamente. A un certo punto, durante la conferenza, al fenomeno finlandese è stato chiesto come pensasse di catturare il mercato della piccola e media impresa. (Tipica risposta torvaldiana: «Personalmente non ho mai provato a catturare nessuno.») Due domande più tardi un giornalista con un'aria da io-sono-l'unico-a-sapere-come-funziona-davvero-l'open-source chiese a Torvalds cosa pensasse delle aziende che registravano i brevetti dei geni agricoli. (Tipica risposta torvaldiana, liberamente citabile: «Penso esistano due tipi di brevetti: quelli abbastanza brutti e quelli davvero brutti.»)

Programmatori, segnatevi questa: se qualcuno inizia a farvi domande sui geni agricoli, probabilmente è ora di procurarvi una segreteria telefonica.

Quindi probabilmente è giusto che Linus non risponda più al telefono. Ma ci mancheranno il candore e la semplicità di Torvalds, che prendevano sempre piacevolmente in contropiede i giornalisti abituati all'aria fritta degli uffici marketing. E speriamo che se quei fax arriveranno veramente sulla sua scrivania e se lui risponderà veramente alle domande, manterrà il suo tono torvaldiano.

Perché se a vincere sarà la voce falsamente amichevole delle PR, l'affare Linux non sarà mai più così divertente.

Ok. Immagino di dovere una spiegazione al signor Berinato. Ma non una scusa.

Chiunque leggesse questo articolo penserebbe che la pressione sempre crescente del mio ruolo di capo-nerd mi abbia trasformato in uno stronzo. Ma non è vero. Io sono sempre stato uno stronzo.

Partiamo dall'inizio. Credo che la segreteria telefonica sia il male. È un esempio perfetto di pessima tecnologia. In effetti è la tecnologia peggiore che esista e io la odio con tutto il cuore. Così alla Transmeta iniziammo con un sistema di posta vocale individuale che permetteva a ogni dipendente di registrare venti minuti di messaggi. Dopo di che, chi chiamava si sentiva dire che la casella vocale era piena e che doveva contattare il centralino. La mia era sempre piena.

Credo siano stati i giornalisti a causare la grande rivolta. Rompevano le scatole alle centraliniste della Transmeta perché la mia casella vocale era piena. Dopo le prime centinaia di volte, le centraliniste iniziarono a irritarsi. Sapevano che a me non interessava la cosa e non volevano avere il ruolo di quelle che mandavano la gente a fare in culo.

Così iniziai a cancellare i messaggi senza ascoltarli, tanto per non far arrabbiare chi stava al centralino. In ogni caso io i messaggi non li ascoltavo quasi mai nemmeno prima. La gente borbotta sempre il proprio numero di telefono e io dovevo ascoltare

quindici volte il messaggio solo per capire cosa stavano dicendo. E poi mi rifiuto di richiamare le persone se non ho un motivo per farlo. La gente aveva questa sensazione piacevole per il fatto di potermi lasciare un messaggio. Finché non capivano che non richiamavo mai.

A quel punto chiamavano il centralino. Lì non sapevano cosa rispondere, così io dissi loro di chiedere a chi chiamava di mandarmi dei fax. I fax sono facili da ignorare quanto i messaggi in segreteria, ma almeno con un fax potevi capire il numero di telefono, nel caso ti interessasse richiamare. A me non interessava mai.

All'inizio i centralinisti dicevano educatamente a chi chiamava di inviare un fax. Alla fine la gente si accorse che io non leggevo i fax, richiamavano nel giro di una settimana e si lamentavano del fatto di avermi già mandato un fax. Così i centralinisti si trovavano ancora presi in mezzo. Gestire le mie chiamate non era il loro lavoro.

La generosa descrizione che il signor Berinato dà di me ai bei vecchi tempi ha preso piede a prescindere dalla realtà dei fatti. Ve lo assicuro: sono sempre stato uno stronzo. Non è una novità.

La soluzione fax non durò a lungo. Alla fine crearono apposta per me un account di messaggistica telefonica senza posta vocale. A quel tempo la Transmeta aveva assunto un addetto alle PR che si offrì volontario per gestire le richieste indirizzate a me. Hanno studiato per questo, mi dicono. Mi dicono anche che dovrei richiamare sempre i giornalisti perché, anche se non voglio parlare con loro, il fatto che li abbia richiamati da loro una sensazione piacevole. La mia reazione è: chi se ne frega delle loro sensazioni?

Ok. Rispondo al telefono quando qualcuno mi chiama mentre sono seduto alla mia scrivania. Ma è una cosa che non dovrebbe essere interpretata come un tentativo di apparire accessibile. E certamente non è una presa di posizione politica. Il punto con l'open source non è mai stato il fatto che io sia più accessibile di chiunque altro. E nemmeno il fatto che io sia più aperto ai suggerimenti altrui. Non è mai stato questo il punto. Il punto è che anche se io fossi il più infame dei demoni infernali, anche se io fossi il Male Puro, la gente potrebbe scegliere di ignorarmi perché può fare tutto da sola. Non sono io a dover essere aperto, sono loro ad avere il potere di ignorarmi. É una cosa importante.

Non esiste una versione «ufficialo di Linux. C'è la mia versione e ci sono quelle di tutti gli altri. Il fatto è che la maggior parte delle persone si fida della mia versione e vi fa conto come fosse *de facto* una versione ufficiale perché mi hanno visto lavorarci sopra per nove anni. Io sono quello che l'ha creato e di solito le persone ritengono che abbia fatto un buon lavoro. Ma mettiamo che io mi rasi la testa e mostri a tutti un tatuaggio con un 666 e dica: «Inchinatevi di fronte a me o vi schiaccerò come vermi!» Mi scoppierebbero tutti a ridere in faccia e direbbero: «OK, prendiamo il kernel e facciamoci ciò che ci sembra giusto

La gente si fida di me. Ma il solo motivo per cui lo fa è perché sinora mi sono dimostrato affidabile.

Questo non vuol dire che abbia voglia di ascoltare i messaggi in segreteria o le telefonate di chiunque si trovi dall'altra parte del filo. Non ho mai pensato che la gente mi dovesse vedere come il bonaccione che risponde a chiunque lo chiami o gli mandi un'e-mail. E già che ci siamo, è strano che ci siano tutti questi articoli in cui

mi dipingono come un monaco o un santo privo di ego a cui non importa nulla del denaro. Nel corso degli anni ho provato a sfatare questo mito, ma i miei sforzi non hanno mai avuto effetto sui giornali. Non voglio essere la persona che la stampa vuole che io sia.

Il fatto è che ho sempre odiato questa immagine da monaco privo di ego. È una cosa da sfigati. È noiosa. Ed è falsa.

X.

Quando sono passato in quattro e quattr'otto dalla mia camera da letto alle luci dalla ribalta ho dovuto imparare molto velocemente dei trucchetti esistenziali che altre persone probabilmente hanno fatto propri ancora prima di andare all'asilo. Per esempio non mi sarei mai immaginato quanto la gente mi avrebbe preso ridicolmente sul serio, qualsiasi cosa facessi. Eccovi due esempi, due variazioni sullo stesso tema.

Ai tempi dell'università avevo un root account sulla mia macchina. Ogni account è associato a un nome. Il nome viene utilizzato a scopo informativo. Così io chiamai il mio root account Linus «God» Torvalds. Ero il Dio di quella macchina, che se ne stava nel mio studio all'università. Niente di che, no?

Ora, se qualcuno vuole vedere chi è collegato a una macchina in Linux o in Unix esegue un *finger.* Con l'avvento dei *firewall, il fingering* non viene più fatto molto spesso. Ma anni fa la gente faceva un finger su un'altra macchina per vedere se l'utente si era collegato o aveva letto la posta. Era anche un modo per controllare il *pla*n di qualcuno, le informazioni personali che aveva pubblicato sulla propria macchina, una specie di predecessore delle pagine web. Il mio plan comprendeva sempre l'ultima versione del kernel. Quindi tenersi aggiornati sulle ultime versioni consisteva nel fare un finger sulla mia macchina. Qualcuno aveva addirittura automatizzato il processo. Facevano un finger su di me ogni ora per stare dietro ai cambiamenti delle versioni di Linux. In ogni caso, quando qualcuno faceva finger su di me, vedeva che il mio root account si chiamava Linus God Torvalds. All'inizio la cosa non costituiva un problema. Poi iniziai a ricevere delle e-mail da gente che mi dava del blasfemo. Così finii per cambiare il nome. Si trattava di persone che si prendevano troppo sul serio, una cosa che mi fa impazzire.

Poi, naturalmente, ci fu l'incidente in North Carolina. Cacchio! Quella sì che fu una brutta storia! Un libro uscito recentemente sulla Red Hat lo fa sembrare un incidente internazionale di proporzioni potenzialmente catastrofiche. In realtà non fu niente di che.

Ero stato invitato a parlare a un incontro di utenti Linux presso la sede della Red Hat a Durham. L'auditorium era strapieno. Quando uscii sul palco tutti si alzarono in piedi e iniziarono ad applaudire. Dissi la prima cosa che mi venne in mente:

«Sono il vostro Dio.»

Una volta per tutte: ERA UNA BATTUTA!

Non era una cosa tipo «Sono davvero convinto di essere il vostro Dio e non dovete mai dimenticarlo. Il senso era: «Ok, Ok, Ok. Lo so che sono il vostro Dio. Ma adesso

sedetevi e risparmiate gli applausi per quando avrete sentito quello che ho da dire, per quanto apprezzi molto la vostra stima preventiva.»

Non posso credere che stia rivivendo tutta questa faccenda di mia spontanea volontà.

Dopo quelle quattro parole tutti restarono in silenzio per un istante. Qualche ora più tardi la mia frase era diventata argomento di discussione nei newsgroup. Lo ammetto: era stata un'uscita di cattivo gusto, ma non lo era stata di proposito. In effetti si trattava probabilmente del mio modo di reagire all'imbarazzo di trovarmi di fronte delle persone che si alzavano in piedi ad applaudirmi solo perché ero arrivato sul palco.

La gente mi prende troppo sul serio. Prendono molte cose troppo sul serio. É una lezione che ho imparato negli anni passati a essere il volto umano di Linux è che c'è qualcosa di ancora peggiore: a certe persone non basta prendere le cose troppo sul serio da sole. Non sono contente finché non convincono altre persone ad abbracciare la loro ossessione. E uno dei motivi per cui non mi piace possedere animali domestici.

Vi siete mai fermati a pensare perché i cani amino tanto gli uomini? No, non è perché i loro padroni li portano ogni sei settimane a fare la toeletta e ogni tanto raccolgono quello che loro lasciano sul marciapiede. È perché ai cani piace che si dica loro cosa fare. Dà loro una ragione per vivere. (Il che è particolarmente importante da quando molti di loro sono stati «licenziata tramite la sterilizzazione, che li ha privati del loro lavoro di riproduttori di nuove generazioni di cani. E con poche eccezioni non c'è grande richiesta per gli altri lavori in cui riescono particolarmente bene, tipo annusare i roditori.) Come umani, voi siete i capi del branco e dite ai cani come si devono comportare. Seguire i vostri ordini è la loro passione. E sono contenti così.

Sfortunatamente anche gli uomini funzionano allo stesso modo. Alla gente piace che ci sia qualcuno a dire loro cosa fare. È il nostro kernel. Tutti gli animali sociali sono fatti così.

Il che non vuol dire essere sottomessi. Vuol dire che è probabile che andiate d'accordo con gli altri quando vi dicono che cosa fare.

Poi ci sono persone con idee individuali, gente che ha delle convinzioni in determinati ambiti, convinzioni forti fino al punto di fare dire loro: «No, io non vi seguo. E queste persone diventano leader. È facile diventare un leader. *(Deve* esserlo, se ci sono riuscito anche io, giusto?) Altre persone che in quegli ambiti non hanno convinzioni tanto forti saranno più che felici di lasciare che questi leader decidano per loro e dicano loro cosa fare.

È un diritto assoluto di qualsiasi essere umano fare ciò che chi si è scelto come leader gli dice di fare. Non discuto questo punto, anche se lo trovo piuttosto deprimente. La cosa che trovo inaccettabile è che qualcuno - che sia un leader o un seguace - provi a imporre agli altri la propria visione del mondo. Il fatto che la gente sia disposta a seguire praticamente chiunque, me compreso, è deprimente. Ed è spaventoso che a quel punto la gente vorrà imporre la propria *seguacit*à - se esiste una parola del genere - agli altri, me compreso.

Non pensate agli evangelizzatori robotizzati e pulitini che bussano alla vostra porta

ogni volta che siete al computer e cercate di concentrarvi su un problema tecnico complesso, o quando i bambini si sono finalmente addormentati e voi state iniziando a fare gli sdolcinati con vostra moglie. Nella comunità open source c'è un esempio più a portata di mano: gli zeloti che ritengono che *ogni* innovazione dovrebbe essere pubblicata in forma di GPL («GPLlata», in gergo hacker). Richard Stallmann vorrebbe che tutto fosse open source. Per lui si tratta di una battaglia politica e vuole usare la GPL come base dell'open source. Non vede alternative. La verità è che io non ho pubblicato Linux in open source per motivazioni tanto elevate. Io volevo un feedback. Ed era così che si facevano le cose agli albori dell'informatica, quando la maggior parte del lavoro veniva svolto nelle università o nei laboratori della difesa e poi finiva per essere estremamente aperto. Quando te le chiedevano, passavi le sorgenti a un'altra università. Ciò che Richard fece, dopo essere stato tagliato fuori da alcuni progetti che amava molto, fu di essere la prima persona a usare consapevolmente l'open source.

Sì, ci sono benefici immensi che derivano dall'aprire le proprie tecnologie e renderle disponibili, come è successo a Linux e a tutta una serie di innovazioni. Per farvi un'idea di questi benefici vi basta considerare gli standard qualitativi relativamente bassi di qualsiasi progetto di software chiuso. La GPL e il modello open source consentono la creazione della tecnologia migliore. Tutto qui. Inoltre impediscono l'accaparramento delle tecnologie e assicurano che chiunque sia interessato a un progetto o a una tecnologia non possa essere escluso dai suoi sviluppi.

Non è un punto da poco. Stallmann, che meriterebbe un monumento per avere creato la GPL, ha fatto nascere il fenomeno del free software soprattutto perché era stato escluso da una serie di progetti interessanti quando questi passarono dal mondo accademico e aperto del Massachusetts Institute of Technology a un ambiente aziendale e chiuso. Il più importante di questi progetti fu quello della macchina LISP Il LISP nacque come parte della comunità delle intelligenze artificiali. Come spesso accade, qualcuno pensò che fosse tanto interessante da dover formare una *start-up,* farne un successo commerciale e guadagnare un sacco di soldi. Nelle università succede sempre così. Ma Richard non faceva parte del gruppo commerciale, così quando il LISP divenne un progetto commerciale con la formazione nel 1981 di una società chiamata Symbolics, ne venne improvvisamente tagliato fuori. Per aggiungere il danno alla beffa, la Symbolics gli rubò di sotto il naso un bel po'di collaboratori del laboratorio di intelligenze artificiali.

La stessa cosa gli accadde in altre occasioni. Per come la vedo io, le sue ragioni per la promozione dell'open source non erano tanto anticommerciali, quanto antiesclusione. Per lui l'open source vuol dire non poter essere tagliati fuori. Vuol dire poter continuare a lavorare a un progetto a prescindere da chi lo mette in commercio.

La GPL è fantastica perché permette a chiunque di entrare (e restare) in gioco. Pensate a che grande progresso per l'umanità! Ma questo vuol dire che tutte le innovazioni dovrebbero essere GPLlate?

Neanche per idea! É il corrispettivo tecnologico del problema dell'aborto. Ogni singolo innovatore dovrebbe poter decidere se GPLlare un progetto o se scegliere un

approccio più convenzionale al copyright. La cosa che mi fa impazzire di Richard è che lui vede tutto in bianco e nero. E questo crea inutili divisioni politiche. Non capisce mai il punto di vista degli altri. Se si interessasse di religione, sarebbe un fondamentalista.

In effetti la cosa più fastidiosa - seconda solo ai fanatici religiosi che bussano alla mia porta per venirmi a dire in cosa dovrei credere - è la gente che bussa alla mia porta (o mi bombarda di e-mail) per dirmi come dovrei pubblicare il mio software. Non dovrebbe essere una questione politica. La gente dovrebbe poter decidere da sola. Una cosa è suggerire a qualcuno di considerare l'idea di GPLlare il proprio software e poi lasciarlo decidere da sé. Un'altra cosa è mettersi a discutere. È davvero terribile quando la gente si lamenta del fatto che io lavori per un'impresa commerciale che non GPLla tutto ciò che produce. Io di solito rispondo che non sono fatti loro.

La cosa che trovo tremendamente irritante in Richard non è il fatto che creda che Linux -poiché il suo kernel faceva affidamento su applicazioni del progetto software gnu - dovrebbe chiamarsi «gnu/Linux». Non è il fatto che ce l'abbia con me perché sono diventato il volto pubblico dell'open source anche se lui condivideva il codice quando io dormivo ancora in un cesto della biancheria. No, il motivo per cui lo trovo tanto fastidioso è che si lamenta costantemente del fatto che altre persone non utilizzano la GPL.

Ammiro Richard (da lontano) per un sacco di motivi. E tendenzialmente rispetto le persone che, come Richard, hanno opinioni morali molto forti. Ma perché non possono tenere per sé queste opinioni? La cosa che più detesto è quando la gente mi dice cosa dovrei o non dovrei fare. Provo una profonda antipatia per le persone che pensano di avere voce in capitolo sulle mie decisioni personali (eccetto, forse, mia moglie).

Nel corso dello sviluppo di Linux sapientoni come Eric Raymond hanno suggerito che il successo del sistema operativo e la longevità della sviluppo open source siano in parte stati un effetto del mio approccio pragmatico e della mia capacità di non schierarmi con nessuno nelle dispute. Sebbene Eric sia probabilmente il migliore teorico del fenomeno open source (e sebbene io sia fermissimamente contrario alle sue idee a favore delle armi da fuoco), credo che la percezione che ha di me sia leggermente sfasata. Non è che io mi astenga dallo schierarmi. È solo che provo un profondo biasimo per chiunque cerchi di imporre la propria morale agli altri. E potete sostituire la parola «morale» con «religione», «fede informatica» o altro ancora.

Se imporre una morale è sbagliato, il passo successivo - istituzionalizzare una morale - lo è doppiamente. Credo fermissimamente nella scelta individuale, il che vuol dire che penso che quando si tratta di questioni morali la scelta spetta a me e a me soltanto.

Voglio decidere da solo. Sono assolutamente contrario a che la società imponga regole non necessarie. Credo fermamente che in privato si dovrebbe poter fare qualsiasi cosa si voglia, purché non si faccia del male a nessuno. Qualsiasi legge che dica il contrario è una legge molto, molto sbagliata. Ed esistono leggi che dicono il contrario. Mi capita di scoprire delle regole che fanno paura, soprattutto tra quelle imposte nelle scuole e ai bambini. Provate a immaginare di imporre delle norme per

insegnare l'evoluzione e di farvi portare da queste norme nella direzione sbagliata. É una cosa spaventosa. È la coscienza sociale che infila il proprio orribile naso in posti con cui non dovrebbe avere nulla a che spartire.

Allo stesso tempo io personalmente credo che esista qualcosa di più importante di me e delle mie decisioni morali individuali: non si tratta della razza umana, ma dell'evoluzione. Di conseguenza voglio che le mie scelte individuali tengano conto delle problematiche sociali. Ma credo sia una cosa innata. Credo che il fatto di tenere conto del sociale sia una cosa innata nella biologia umana, nell'evoluzione. In caso contrario saremmo scomparsi molto, molto tempo fa.

L'unica altra cosa contro cui valga la pena di concionare: la gente che tende a fare delle prediche. Non c'è alcun motivo valido per evangelizzare a destra e a manca e per sentirsi tanto nel giusto mentre lo si fa.

E io ora sto parlando come uno di loro.

Ma è una trappola in cui è facile cadere quando la gente inizia a prenderti troppo sul serio.

XI.

Gli americani se la menano un sacco con il 17 di marzo (il giorno di san Patrizio), il 5 di maggio (Cinco de Mayo) e il 12 ottobre (il Columbus Day), mentre non prestano grande attenzione al 6 dicembre, che come potrebbe dirvi qualsiasi abitante di Helsinki è il giorno dell'Indipendenza finlandese.

La maggior parte dei finlandesi festeggia il giorno dell'Indipendenza come festeggia qualsiasi altra cosa: dandosi alle gozzoviglie. Si danno alle gozzoviglie - anche secondo gli standard finlandesi - la sera prima e si riprendono davanti al televisore per quasi tutto il giorno della festa nazionale. L'unica altra possibilità sarebbe uscire e farsi passare il doposbronza nella neve.

La cosa che tiene tutti incollati alla TV è un unico evento: il Ballo del Presidente. In Finlandia non è che ci siano grandi eventi in termini di alta società, e il Ballo del Presidente è fondamentalmente l'unico evento mondano super-chic dell'anno. Viene trasmesso a diffusione nazionale per impedire di guidare alla gente ancora mezza ubriaca e perché possiamo dimostrare a noi stessi di saper mettere in scena la nostra dignitosa versione della Notte degli Oscar. Anzi, per usare una metafora migliore: il Ballo del Presidente è il Superbowl dell'alta società finlandese.

Così per tutta quella giornata, da Utsjoki nel nord fino ad Hanko nel sud, i finlandesi masticano gravlax e aspirina mentre guardano una processione di invitati - uomini in frac e donne con abiti da sera oltraggiosi (almeno per la Scandinavia) - che stringono la mano al presidente.

Il millenovecentonovantanove fu l'anno in cui mi ci invitarono.

Se sei un ambasciatore finlandese o se fai parte del parlamento finlandese vieni invitato automaticamente. Altre cento o duecento persone vengono invitate per i più svariati motivi. Magari hanno vinto una medaglia alle Olimpiadi, oppure hanno contribuito alla campagna presidenziale. Se sei il capitano della squadra di hockey su

ghiaccio che ha appena vinto il campionato del mondo, vieni invitato. Se il sistema operativo che hai creato è diventato famoso in tutto il mondo, vieni invitato. Insieme a tua moglie.

In effetti fu per pura fortuna che io e Tove ci potemmo andare. Ad agosto avevamo chiesto all'INS il permesso di fare un viaggio andata e ritorno in Finlandia. Fino all'inizio di novembre non ricevemmo l'approvazione. Due settimane più tardi fummo invitati al Ballo del Presidente.

Provate a immaginarvelo. Immaginatevi 2000 finlandesi - e nemmeno i 2000 finlandesi più importanti - ammassati dentro il castello presidenziale. Quell'edificio era stato costruito da un mercante russo e in realtà si tratta semplicemente di una casa molto grande, non proprio una villetta unifamiliare, piuttosto una casa per una famiglia che ha un sacco di personale alle proprie dipendenze: cuochi, cameriere e roba del genere. Non è un posto grandissimo.

Ci arrivi. Qualcuno ti prende il cappotto e a quel punto sei incastrato. Non sai dove andare. Zuppiere da punch dappertutto. Ovviamente sono piene di vodka. Non saremmo in Finlandia, altrimenti. Ti ci vuole un po'per trovare qualcuno con cui parlare. Finisci per chiacchierare con i giornalisti perché, sinceramente, sono i più interessanti tra i presenti. (Forse era il punch a farli sembrare più interessanti di un parlamentare di... chessò... Lahti.)

Non mi aspettavo di divertirmi un granché, perché non conoscevo molte persone. Non avevano invitato altra gente dell'open source. Pensavo potesse essere una cosa tipo il militare, di quelle che sono più divertenti dopo, quando sono finite. E invece mi divertii.

Tove indossava un abito verde che avrebbe fatto scalpore e attratto l'attenzione dei media anche se fossimo stati agli Oscar e non al Ballo del Presidente finlandese. Era bellissima, la Finlandia quell'anno non aveva vinto il campionato del mondo di hockey su ghiaccio, e quindi la stampa ci nominò Re e Regina del Ballo.

Che ci volete fare?

«Tu in questa casa ci entri da amico, non da giornalista. In questa casa i giornalisti non possono entrare.»

Non avevo mai visto Tove tanto agitata. Mi aveva accolto sulla porta della loro nuova casa il giorno stesso in cui lei e Linus avevano avuto le chiavi. È una di quelle case elefantiache alla californiana: la Stanza dei Divertimenti (che ora ospita il tavolo da biliardo di Linus) probabilmente non ha nemmeno lo stesso codice d'avviamento postale della Stanza Super Bonus, dove ora stanno dormendo Patricia e Daniela, anche se potrebbe accogliere agevolmente un intero asilo. Dalla porta d'ingresso parte un lungo e ampio corridoio che, svoltati alcuni angoli, porta fino alla sala. Se togliessero le eleganti piastrelle italiane sarebbe un posto fantastico dove le ragazze un giorno potrebbero esercitarsi con lo skateboard. Lo studio di Linus al primo piano ha come porta una vetrata a specchio scorrevole. Cinque bagni. Forse quando leggerete questo libro ne avranno trovato qualcun altro disperso per le ali più remote della casa. Il tutto in un quartiere residenziale lontano dal cuore della Silicon Volley.

Nicke Torvalds è qui in visita. Padre e figlio tornano da un giro alla vecchia casa in un BMW z3 a noleggio. È il modello che Linus sta per comprarsi e Nicke questo pomeriggio la userà per andare alla biblioteca della Stanford University. Ma prima si appoggia alla vasca calda posizionata nel giardino dall'aria selvaggia e annuncia che questa è la casa più grande mai posseduta da chiunque abbia portato il cognome Torvalds. Poi prende un pezzo di carta ed elenca tutti e venti i Torvalds. Non sapeva che ce n'era un ventunesimo in arrivo.

Anche Linus è elettrizzato dalla casa vuota. Nicke fa un ripresa video del circondario e io chiedo a Linus di portare Tove in braccio oltre la porta in modo che possa fotografare l'evento. Tutto quel mostrare pubblicamente il proprio affetto è divertentissimo e molto poco finlandese.

«Avresti mai pensato che la nostra casa sarebbe stata così grande», mi chiede Tove.

***

Tove doveva andare all'apertura dell'IKEA di Emeryville per comprare degli armadi per la nuova casa, così suggerii a Linus di portare le bambine in una casa che avevo in affitto a Stinson Beach. Quando arrivarono dissi a Linus di provare il kayak nella laguna. Se ne andò a pagaiare da solo, poi con ciascuna delle ragazze. Quando tornò al molo aveva i pantaloni fradici.

Volevo che Linus mi desse il suo parere su un capitolo che avevo intitolato «Il successo mi rovinerà?» e così portai le ragazze in spiaggia perché potesse leggerlo indisturbato. Patricia e Daniela passarono una mezz'ora a cercare stelle marine e a ciabattare nell'acqua, dopodiché una delle due mi disse «Kisin kommer», che si potrebbe tradurre più o meno con: «Mi scappa la cacca».

Tornammo a casa e trovammo Linus seduto al computer in mutande, accanto a un sacchetto di salatini, tutto intento a scrivere qualcosa. Gli ci vollero almeno quindici secondi per accorgersi che eravamo arrivati. Alzò lo sguardo dal monitor. Le sue prime parole furono: «Amico, il tuo Macintosh fa schifo.»

Poi: «Ah, ho messo i pantaloni in lavatrice.»

Aveva cambiato il titolo del capitolo in «Fama e ricchezza»: gli sembrava che «Il successo mi rovinerà?» fosse troppo egocentrico. Voleva altro tempo per scrivere, così portai le ragazze in spiaggia finché Linus non ebbe finito il capitolo.

XII.

È facile combattere i mulini a vento se non sai quanto è difficile.

Cinque anni fa tutti avevano un tono estremamente dubbioso quando mi chiedevano se pensavo che Linux sarebbe riuscito a conquistare il mercato desktop e intaccare il predominio di Microsoft. Io rispondevo invariabilmente di sì. Loro assumevano un'aria scettica. Il fatto è che probabilmente conoscevano la realtà meglio di me.

Io non capivo davvero tutti i passaggi necessari per raggiungere quell'obiettivo. Non sapevo cosa servisse per risolvere i problemi tecnici relativi allo sviluppo di un sistema operativo robusto e portabile, e ancor meno cosa servisse per fare di questo sistema operativo un successo commerciale, oltre che tecnico. Se avessi saputo in anticipo quante infrastrutture si sarebbero rese necessarie perché Linux avesse il successo che ha avuto, mi sarei scoraggiato. Non basta essere bravi. Bisogna esserlo, certo, ma poi le cose devono anche andare tutte per il verso giusto.

Qualsiasi persona sana di mente avrebbe dato uno sguardo alla montagna che aveva deciso di scalare e ci avrebbe subito rinunciato. Pensate solo ai problemi tecnici per supportare i PC, che quanto ad hardware sono la cosa più variabile del mondo. Devi dare una mano a gente che ha dei bachi che non puoi riprodurre su applicazioni di cui non ti importa nulla. Ma ti importa di Linux, e allora ti importa anche di cercare di sistemare questi bachi.

Anche solo per pensare di entrare nel mercato commerciale, devi offrire un livello rispettabile di assistenza agli utenti. Sin dall'inizio con Linux potevi avere una vera assistenza realizzandola all'interno dell'azienda. Ma per realizzarla in grande stile dovevi avere un sacco di persone e un sacco di infrastrutture. Non basta mettere su un numero verde per i primi trenta giorni. Oggi l'assistenza non è più un problema, almeno in parte, perché la puoi comprare da un sacco di società diverse: Linuxcare, Red Hat, IBM, Silicon Graphics, Compaq, Dell. Ma all'inizio era una cosa che andava assolutamente creata. Per un sacco di tempo non ci avevo nemmeno pensato. È stata per anni una grossa sfida.

A differenza degli uomini d'affari con una buona preparazione tecnica e dei giornalisti con interessi commerciali, io ero uno sviluppatore di software puro, e non avevo un'idea di tutto quello che sarebbe stato necessario. Mi sarebbe bastato riflettere sui problemi tecnici per rinunciare a imbarcarmi in questa impresa. Se avessi saputo quanto lavoro sarebbe stato necessario, e che a dieci anni di distanza sarei ancora stato qui a occuparmene (e che per tutti questi dieci anni sarebbe stato praticamente un lavoro a tempo pieno) non avrei mai neppure iniziato.

E gli insulti! Oggi non ne ricevo più tanti, ma a volte succede ancora. Gente a cui non piace l'open source o gente incazzata per dei bachi mi manda delle e-mail in cui mi maledicono per le loro frustrazioni. In confronto alla quantità di e-mail positive che ricevo, è poca cosa. Ma non è che non mi pesi.

Sì, se avessi saputo quanto avrei dovuto lavorare duro, quanto sarebbero state

difficili un sacco di cose, probabilmente non l'avrei fatto. Se avessi avuto abbastanza cognizione di causa per immaginare i problemi in anticipo, probabilmente non avrei mai portato Linux oltre la sua prima release. Se avessi saputo quanti particolari si devono aggiustare e quanto la gente si aspetta da un sistema operativo, avrei anche immaginato gli scenari da film dell'orrore generati da tutte le cose che non sarei riuscito a gestire.

Ma non potevo immaginare nemmeno gli aspetti positivi della cosa. Tutto il supporto che ho avuto, tutte le persone che hanno collaborato a questo progetto. Ok, lo ammetto. Se avessi immaginato gli aspetti positivi, probabilmente *l'avrei* fatto.

Proprietà intellettuale

Il dibattito sulla proprietà intellettuale di questi tempi è tanto acceso che non mi riesce di entrare in un qualsiasi bagno pubblico senza imbattermi in una scritta murale a favore dell'uno o dell'altro schieramento. Qualcuno pensa che i brevetti e le altre leggi sulla proprietà intellettuale siano la rovina dell'universo libero, che queste leggi non siano solo scorrette ma addirittura *malvagie,* e che dovrebbero essere cancellate il prima possibile. Altri sono convinti che buona parte dell'economia mondiale sia fondata sulla proprietà intellettuale. E *queste* persone farebbero qualsiasi cosa per rafforzare lo stato legale dei diritti di proprietà intellettuale.

La conseguenza è che le scritte murali sull'argomento a volte usano un linguaggio abbastanza esplicito.

Naturalmente la maggior parte di queste scritte si trova nei bagni pubblici virtuali di internet, più che in quelli dei locali più alla moda di San Jose[10] . In rete c'è gente che non fa altro che litigare su un sacco di argomenti collegati alla proprietà intellettuale, dal Primo Emendamento al fatto che la legge sulla proprietà intellettuale possa o meno rendere impossibile uno sviluppo futuro dell'open source.

Per quanto mi riguarda mi sento decisamente schizofrenico in proposito.

Non è che non abbia una mia opinione: ho delle opinioni molto forti sul valore della proprietà intellettuale, ma appartengono in qualche modo a entrambi gli schieramenti. Ve lo assicuro: è una cosa che può confondervi le idee. E così finisco per litigare con entrambi gli schieramenti. Penso che questo accada perché esistono due diversi aspetti della proprietà intellettuale, e in comune hanno solo il nome.

Per molte persone, me compreso, la proprietà intellettuale è qualcosa che riguarda l'inventiva umana, ciò che ci rende *umani* anziché animali (oltre ai pollici opponibili, naturalmente). In quest'ottica il termine «proprietà intellettuale» è già in sé un affronto: non si tratta di una proprietà da vendere come una merce qualsiasi, è l'atto della creazione, è la cosa più grande che un uomo possa fare. È Arte con l'A maiuscola. È la Monna Lisa, ma è anche il risultato finale di una lunga notte di programmazione, un risultato finale di cui come programmatore sei maledettamente *fiero.* È qualcosa di tanto prezioso che non è nemmeno possibile venderlo: è una parte di te, per sempre.

Questo genere di creatività - che si tratti di pittura, musica, scultura, scrittura o programmazione - dovrebbe essere sacra. Il creatore e la cosa che ha creato hanno un legame che non può essere reciso. É come il legame tra madre e figlio, o tra il cibo cinese da quattro soldi e l'odore di fritto. Ma allo stesso tempo è qualcosa di cui chiunque, ovunque si trovi, dovrebbe poter entrare a far parte, perché è dell'umanità.

All'angolo opposto, con un peso di circa sette godzliardi di dollari americani

---

[10] Questa, come potrà dirvi chiunque viva a San Jose, si chiama ironia. A San Jose non ci sono locali alla moda. Quelli che ci vivono, se si vogliono divertire, salgono in macchina e se ne vanno a San Mateo.

all'anno, la proprietà intellettuale è un grande business. Hanno messo un'etichetta con il prezzo alla creatività umana, e anche bella salata. La creatività è merce rara e quindi non è solo costosa, ma anche estremamente redditizia. Il che porta in campo un tipo di argomentazioni e di persone completamente differenti. Quel genere di persone che chiama «proprietà» il risultato finale della creatività umana. Per non parlare naturalmente degli avvocati.

Rileggete il titolo di questo capitolo. Le persone che sostengono la «proprietà» stanno vincendo. In fondo, il loro nome ha avuto il sopravvento. E allora qual è il problema?

L'esempio più conosciuto di proprietà intellettuale è l'idea di diritto d'autore, il copyright. I copyright sono fondamentalmente la codifica del diritto di qualsiasi creatore di fare ciò che vuole della propria creazione. Il «proprietario della creazione può decidere come questa debba essere utilizzata.

Il copyright, da un punto di vista legale, è anche molto facile da ottenere. Non c'è bisogno di registrarlo: detieni automaticamente il copyright su qualsiasi lavoro creativo tu produca. Si tratta di una distinzione importante, che differenzia il copyright dalla maggior parte delle altre leggi relative alla proprietà intellettuale, soprattutto perché rende di fatto semplice per gli individui - e non solo per le grandi aziende - possedere dei copyright. Puoi possedere un copyright semplicemente perché hai scritto, dipinto o comunque creato qualcosa di unico. Se lo desideri puoi aggiungervi una legenda del tipo «© Copyright 2000 Pinco Pallino, ma in effetti non hai alcun bisogno di farlo. Sei il proprietario del copyright, che tu lo dichiari o no. Dirlo rende solo più semplice rintracciarti per chiunque voglia utilizzare il tuo lavoro.

Naturalmente il mero possesso del copyright non è particolarmente utile. Ma il fatto di possedere ciò che create implica che possiate decidere come usarlo. Potete concederlo in licenza. E ancora *meglio* che venderlo: anziché vendere la vostra opera, potete vendere la licenza di farne determinate cose, continuando a conservarne il copyright. In altre parole: potete avere la botte piena *e* la moglie ubriaca. È così che vengono create le varie Microsoft del mondo, vendendo all'infinito il diritto di utilizzare qualcosa, senza perdere nulla. Non c'è da meravigliarsi del fatto che la gente *adori* possedere questo tipo di proprietà.

Non iniziate a sentire puzza di bruciato? Se in tutto ciò non vedete niente di strano, avrei un ponte e qualche tratto di spiaggia da vendervi...

Quello che probabilmente subodorate è il problema fondamentale della proprietà intellettuale: voi, come possessore della proprietà intellettuale, potete effettivamente venderla per sempre, senza perdere nulla. Non rischiate niente e di fatto potreste decidere di scrivere la vostra licenza in modo che dica che anche se la proprietà risultasse *difettosa,* voi non potete essere ritenuti in alcun modo responsabili. Vi sembra assurdo? Io non ci scommetterei.

Difetto: nessuna protezione per l'utente.

Ma le cose possono peggiorare. Il possessore del copyright non ha solo il diritto di vendere la sua proprietà senza perderla, ma ha anche quello di fare causa a chiunque venda una proprietà che *assomigli* alla sua. Chiaramente il possessore del copyright ha dei diritti su *quell'opera derivata.*

Chiaramente? Andiamoci piano. Dove sta la differenza tra ispirarsi e copiare? E

cosa succede quando persone differenti se ne escono con idee simili? Chi si assicura la gallina dalle uova d'oro, la possibilità di continuare a vendere la propria idea, nonché il diritto di dire agli altri di levarsi di torno? Non sono solo i consumatori a non essere protetti: nemmeno *le altre* persone creative sono protette dalla nozione di «proprietà intellettuale».

La cosa che a questo punto rende difficile la discussione è che molte argomentazioni a favore di una proprietà intellettuale più forte sono basate sull'idea di dare una maggiore «protezione» agli inventori e agli artisti. La cosa che molti sembrano non capire è che dare diritti tanto forti ad alcune persone vuol dire inevitabilmente toglierli ad altre.

E forse non vi sorprenderà scoprire che i propugnatori di leggi più rigide sulla proprietà intellettuale sono le entità che ne traggono un maggiore profitto economico. Non gli artisti o gli inventori, ma i distributori della proprietà intellettuale, ovvero le aziende che campano sulla creatività altrui. Ah, e gli avvocati, naturalmente. Il risultato finale? Gli emendamenti alle leggi sul copyright, per esempio il tristemente famoso Digital Millennium Copyright Act (DM-CA), che elimina le ultime vestigia dei diritti del consumatore nell'uso di materiali protetti da copyright.

Ora, se vi state facendo l'idea che io pensi che il copyright sia negativo in sé, vi sbagliate. Si dà il caso che io adori il copyright. È solo che non credo sia giusto spingere troppo in là i diritti degli autori. Non fino al punto di fregare il consumatore. E non lo dico solo in quanto consumatore, ma anche come produttore di materiale protetto da copyright, sia nella forma di questo libro sia come Linux.

Io, come proprietario di copyright, ho i miei diritti. Ma con i diritti vengono anche i doveri, ovvero - come dicono in quartieri alti - *noblesse oblige*. E così io ho il dovere di utilizzare questi diritti in modo responsabile e non come arma contro altri che non godono degli stessi diritti. Come ha detto una volta - parola più parola meno - un grande americano: «Non chiedetevi cosa può fare il vostro copyright per voi, chiedetevi cosa potete fare voi per il vostro copyright».

E alla fine il copyright, nonostante il DMCA, è una forma di proprietà intellettuale abbastanza moderata e ben congegnata. Continua a esistere l'idea di «giusto utilizzo e il fatto di possedere un copyright non dà al proprietario *tutti* i diritti sull'opera.

Non si può dire lo stesso dei brevetti, dei trademark e dei segreti aziendali, le droghe pesanti della proprietà intellettuale. In particolare le discussioni tra gli addetti ai lavori sui brevetti dei software sono tanto infiammate da essere incluse ufficialmente tra gli argomenti che è meglio non affrontare in occasioni formali, insieme al controllo delle armi da fuoco, alle leggi sull'aborto, alla legalizzazione della marijuana e al problema se sia più buona la Pepsi o la Coca Cola. Il motivo è che questi brevetti per molti aspetti danno un controllo sulle nuove invenzioni simile a quello del copyright, conservandone però pochi degli aspetti positivi.

Uno dei problemi principali dei brevetti è che, a differenza dei copyright, non si ottiene un brevetto per il solo fatto di avere inventato qualcosa. No, per ottenere un brevetto devi passare attraverso un laborioso e interminabile processo burocratico presso l'ufficio brevetti. Fare la coda all'ufficio brevetti tra l'altro è un po'come mettersi in fila davanti al DMV, solo che devi sapere che in coda con te ci sono almeno dodici avvocati specializzati in brevetti e che la fila dura DUE ANNI. In

poche parole, non è una cosa che fai il venerdì sera tanto per divertirti se i bambini si addormentano presto.

Per aggiungere il danno alla beffa, l'ufficio brevetti non dispone necessariamente delle risorse per verificare se il tuo nuovo brevetto sia davvero una cosa innovativa. Non è come se avessero degli Einstein[11] a lavorare per loro, per cui risulta abbastanza difficile valutare in modo appropriato le nuove invenzioni. E questo significa che in molti casi sono state brevettate delle vere e proprie bufale.

Qual è il risultato? Sono molto pochi i singoli individui che ottengono dei brevetti, per ovvie ragioni. Le *aziende,* invece ne ottengono a tonnellate. Li usano come armi contro le altre società che minacciano di denunciarle per brevetti che *loro* possiedono. Il sistema attuale dei brevetti è fondamentalmente una Guerra Fredda basata sulla proprietà intellettuale anziché sulle armi nucleari. E non è molto più simpatica di quella originale. I singoli inventori vengono lasciati a prender polvere nei rifugi antiatomici: si trovano ad avere a che fare con un sistema impazzito e non dispongono delle risorse per assumere dodicimila avvocati.

Ora, se non volete essere costretti a occuparvi dei brevetti, potete darvi alla droga pesante della proprietà intellettuale: il segreto aziendale. Il vantaggio del segreto aziendale è che non vi dovete preoccupare di un Ufficio Segreti Aziendali, perché non esiste niente del genere. Vi basta mettere sulla vostra proprietà individuale un timbro che dice «SEGRETO» ed è fatta. Potete sempre parlarne con altre persone, ma dovete dire loro che è un segreto.

In passato la gente faceva spesso così ed è per questo che sono state introdotte le leggi sui brevetti. Per incoraggiare i singoli e le aziende a divulgare i propri segreti, la legge sui brevetti ti dava una protezione sul mercato per qualche tempo *se divulgavi il tuo segreto per il successo.* Una semplice forma di dare-per-avere: tu dici come fai una cosa e noi li diamo i diritti esclusivi per tot anni.

Prima dei brevetti la gente proteggeva gelosamente i propri vantaggi tecnologici e si portava i segreti nella tomba. Questo era evidentemente un male per l'evoluzione tecnologica, perché tecnologie promettenti non venivano mai divulgate. La promessa dei diritti esclusivi fece dei brevetti un incentivo molto forte a rivelare tutto, perché non ti dovevi più preoccupare che la concorrenza scoprisse cosa stavi facendo.

Ma stiamo parlando del passato. Cosa succede oggi? Oggi anche i segreti aziendali hanno una protezione legale, per motivi che non riusciamo proprio a immaginare. Qualsiasi persona sana di mente può capire che una volta rivelato, un segreto non è più tale. Se non nei misteriosi e tortuosi meandri delle leggi sulla proprietà intellettuale, in cui i segreti continuano a essere tali anche quando tutti ne sono venuti a conoscenza. É un mondo in cui le conoscenze che hai in testa possono fare in modo che ti facciano causa, se vai a lavorare per l'azienda sbagliata. Alcune leggi sulla proprietà intellettuale sono assolutamente spaventose.

L'open source è in larga misura un modo per trovare pace in questa guerra sulle proprietà intellettuali. Sebbene molte persone abbiano delle proprie opinioni su ciò che l'open source cerchi di fare veramente, per molti versi lo potete considerare una

[11] In effetti Einstein, mentre elavorava la teoria speciale della relatività, lavorava proprio per l'ufficio brevetti. Ma lui era speciale. Lo ammette anche la maggior parte dei dipendenti dell'ufficio brevetti

sorta di distensione hi-tech, un modo per disinnescare l'arma del copyright in questa lotta per la proprietà intellettuale.

Quindi l'open source utilizzerebbe l'arma legale del copyright come invito a entrare in gioco, piuttosto che come arma contro gli altri. È sempre lo stesso vecchio mantra - Fate l'amore, non fate la guerra - solo a un livello un po'più astratto (o forse *molto* più astratto, considerati alcuni *geek* di mia conoscenza).

Ma come in tutti i grandi dibattiti filosofici, c'è sempre l'altra faccia della medaglia. È qui che entra in gioco la mia schizofrenia.

Ho cercato di spiegare perché molte persone ritengano che la proprietà intellettuale, e soprattutto il rafforzamento delle leggi che la riguardano, sia il Male puro. Buona parte della comunità open source (e non solo, per dirla tutta) non vorrebbe altro che la distruzione di tutte le armi nucleari e l'abolizione totale della Guerra Fredda della conoscenza. Ma non tutti sono d'accordo.

L'altro lato della medaglia è che la proprietà intellettuale magari è ingiusta e le leggi sulla proprietà intellettuale forse sono state concepite soprattutto per avvantaggiare le grandi aziende rispetto ai consumatori e ai singoli autori e inventori, ma ci si può fare *un sacco di soldi!* È una cosa che concentra il potere nelle mani di pochi, e il fatto stesso di essere un'arma tanto potente la rende assai efficace sul mercato. E lo stesso motivo che faceva delle armi nucleari la forza di base della Guerra Fredda a rendere la proprietà intellettuale tanto attraente nella guerra delle tecnologie. E le tecnologie vendono.

Per non parlare del circolo virtuoso che generano. Dato che la proprietà intellettuale è un'ottima fonte di entrate, vengono spesi molti soldi per creare *pi*ù proprietà intellettuali. E questo è un fatto molto importante. Le guerre sono sempre state storicamente una fonte di invenzioni e di grandi progressi ingegneristici (all'inizio anche il computer fu sviluppato soprattutto a scopi militari): allo stesso modo anche la guerra virtuale sui diritti della proprietà intellettuale è un potente propellente per la macchina che tiene in moto lo sviluppo tecnologico. E questa è una buona cosa.

Naturalmente io, essendo uno snob intellettuale, sono convinto che la semplice messa in campo di risorse non serva necessariamente a generare una vera creatività. Pensate al settore discografico attuale. Si spendono ogni anno gargantiliardi di dollari per trovare un nuovo artista alla moda, ma nessuno pensa che le Spice Girls (che sono state riccamente ricompensate per il loro contributo all'arte) possano essere confrontate con Wolfgang Amadeus Mozart (che è morto povero in canna). Per cui investire dei soldi su un problema non serve per creare dei geni.

Ma lo snobismo intellettuale - la filosofia «i-geni-non-si-comprano-al-supermercato» - non funziona come modello commerciale a lungo termine. I flussi creativi sono tanto imprevedibili, così difficili da scovare, che un piano a lungo termine non si può basare sulla premessa di trovare un genio. Lo sviluppo tecnologico odierno (e purtroppo anche la musica) non dipende dagli Einstein (o dai Mozart) ma da un grande esercito di ingegneri sgobboni (e nel caso della musica da giovani donne ben dotate) che solo occasionalmente lasciano intravedere brevi lampi di genio, Le risorse aggiuntive non creano una grande arte, ma un progresso lento e costante. E alla fine questa è la soluzione migliore.

L'idea degli ingegneri sgobboni può avere una minore attrattiva romantica rispetto a quella del genio eccentrico. Pensate a quanti film sono stati fatti sugli Scienziati Pazzi e a quanti pochi ne avete visti sugli Ingegneri Sgobboni. Però quando si parla di affari gli occasionali lampi di genio vi fanno comodo, ma vi serve molto di più un flusso di progressi costanti nel tempo.

È qui che la proprietà intellettuale si mostra in tutta la sua potenza: grazie alla sua alta redditività, è stato trasformato in un Sacro Graal che nelle moderne aziende tecnologiche nutre la grande macchina. E così il processo continua grazie alle protezioni della proprietà intellettuale. Non sarà più un processo molto creativo, ma è affidabile.

Vedo quindi entrambi i lati della medaglia, anche se devo ammettere che preferirei quasi sempre che il mondo della tecnologia fosse più divertente e più ispirato. Un mondo in cui i fattori economici non dovessero *sempre* prevalere. Io ho un sogno: un giorno le leggi sulla proprietà intellettuale saranno dettate dalla morale e non da chi si prende la fetta più grossa della torta.

Credetemi, capisco i problemi economici. Allo stesso tempo però non posso fare a meno di desiderare che non avessero un impatto negativo tanto forte sulle attuali leggi sulla proprietà intellettuale. Gli incentivi economici a rafforzare il possesso delle proprietà intellettuali e la difficoltà a esprimere concetti come «giusto utilizzo e «morale» in un testo legale hanno causato l'allontanamento progressivo dei due punti di vista. Come in una disputa tra vicini di casa, nessuna delle due parti è disposta a riconoscere che la soluzione giusta sta probabilmente in un qualche punto tra i due estremi.

Chiaramente, come dimostra la sfortunata approvazione del DMCA, gli incentivi economici vanno alla grande. La domanda è: quale tipo di legge sulla proprietà intellettuale contribuirebbe allo sviluppo, pur essendo al tempo stesso meno pilotata dai più crassi interessi economici?

Il problema è ingigantito dal fatto che la tecnologia moderna (e in particolare internet) sta indebolendo molte forme tradizionali di protezione della proprietà intellettuale. Lo sta facendo tanto velocemente che è difficile tenere il passo, e in modi che nessuno avrebbe potuto prevedere. Chi avrebbe immaginato che le nonnine del Midwest si sarebbero messe a piratare in internet le istruzioni sul punto croce? La possibilità di copiare opere d'ingegno - e la tecnologia stessa - su ampia scala è ormai tanto diffusa e disponibile che le istituzioni con proprietà intellettuali di grande valore stanno correndo a destra e a manca per mettere al riparo i propri interessi. Stanno facendo il possibile per rendere illegali queste copiature e per introdurre nuove misure che mettano fuori legge le tecnologie che possono essere utilizzate per la pirateria.

Cosa c'è di sbagliato? Il problema è che molti tentativi di rendere più difficile l'uso illegale di una proprietà intellettuale rendono molto più difficile anche un suo uso legale. Il classico esempio di tutto questo nel mondo Linux è il cosiddetto caso giudiziario DeCSS.

Di cosa si trattava? Alcune persone che lavoravano a una tecnologia per decodificare i film in DVD vennero denunciati dall'industria del divertimento per aver messo il codice a disposizione di altri su internet. Al giudice che si occupava del

caso non importava che lo scopo del progetto fosse perfettamente legale; il fatto che potesse *potenzialmente* essere utilizzato a scopi illegali rendeva illegale negli Stati Uniti la diffusione anche solo delle informazioni su dove trovare le istruzioni per la decodifica. Il nome «DeCSS» deriva dal fatto che il progetto lavorava all'aggiramento del Content Scrambling System dei DVD, il cosiddetto CSS: si «de-CSSava» qualcosa per rimuovere la protezione e poter guardare i film sul proprio computer.

Questo è un perfetto esempio in cui le leggi sulla proprietà intellettuale vengono utilizzate non per incrementare lo sviluppo delle innovazioni, ma per controllare il mercato, per controllare ciò che i consumatori possono o non possono fare. É un esempio di legge sulla proprietà intellettuale utilizzata in modo sbagliato.

Questi utilizzi sbagliati del potere della proprietà intellettuale peraltro non si limitano alle tecnologie. Un altro esempio classico è l'uso della legge sui segreti aziendali per perseguire e perseguitare le persone che hanno cercato di far arrivare al pubblico delle informazioni su Scientology. La Chiesa di Scientology è riuscita a sostenere che le loro scritture («Tecnologie Avanzato) ricadevano nelle categorie protette dal segreto aziendale e hanno usato la leggi sulla proprietà intellettuale perché non fossero rese pubbliche.

Quali sono le alternative? Immaginate una legge sulla proprietà intellettuale che tenesse davvero conto dei diritti delle *altre* persone. Immaginate leggi sulla proprietà intellettuale che incoraggino l'apertura e la condivisione. Leggi che dicano: certo, potete sempre avere dei segreti, che siano tecnologici o religiosi, ma questo non rende *obbligatoria* la protezione legale di questi segreti.

Sì, lo so. Sono un sognatore.

La fine del controllo

Il modo per sopravvivere e avere successo è fare il prodotto dannatamente migliore che potete. E se con questo non riuscite a sopravvivere e avere successo, vuol dire che doveva andare così. Se non sapete fare una buona auto, meritate di andare a fondo come un sasso, come accadde all'industria automobilistica americana negli anni Settanta. Il successo dipende dalla qualità del progetto e dal fatto di dare alla gente ciò che vuole.

Non dal cercare di controllare le persone.

Il problema è che le persone e le aziende sono troppo spesso spinte dalla pura cupidigia. E questo, sul lungo periodo, le porta sempre alla disfatta. La cupidigia porta a decisioni governate dalla paranoia e a un bisogno di controllo totale. Si tratta di decisioni sbagliate, miopi, che portano al disastro, o quasi. Un semplice esempio che tutti hanno presente è quello del fenomenale successo iniziale della tecnologia wireless in Europa, a discapito delle aziende americane. Mentre le aziende statunitensi cercavano di controllare individualmente il mercato utilizzando standard proprietari, le società europee si accordarono per uno standard unico, il GSM, e scelsero di competere sulla base di quale azienda sapesse creare il prodotto migliore e fornire il miglior servizio. Le aziende americane sono cadute una per una, sconfitte dai loro stessi standard in competizione. Con un mercato unificato da uno standard comune, le società europee hanno avuto tutte quante un boom. E per questo che i ragazzi a Praga si scambiavano messaggi SMS anni prima che quelli di Peoria avessero mai anche solo sentito parlare di questa nuova possibilità di copiare nei compiti in classe.

Se provate a fare i soldi controllando una risorsa, finirete per trovarvi gambe all'aria. È una forma di dispotismo e la storia è piena di esempi dei suoi pessimi effetti. Mettiamo di essere nel West americano in pieno Ottocento e di controllare le sorgenti d'acqua per i contadini della zona. La forniamo goccia a goccia e la facciamo pagare moltissimo. A un certo punto diventa inevitabilmente redditizio per qualcun altro trovare un modo per far arrivare l'acqua da qualche altra parte, e il nostro mercato crolla. Oppure la tecnologia progredisce, l'acqua può essere incanalata in tubi e fatta arrivare da lontano. In ogni caso le circostanze cambiano, noi perdiamo la nostra posizione di forza e ci ritroviamo con in mano un pugno di mosche. Succede continuamente ed è affascinante come la gente ancora oggi non abbia imparato a prevederlo.

Un salto in avanti: il settore discografico negli ultimi anni del XX secolo. La risorsa che controlla è il divertimento. Un'azienda possiede i diritti sull'opera di un artista. Quell'artista produce una serie di singoli di successo, ma l'azienda mette al massimo uno o due di questi singoli su ogni CD che produce. In questo modo possono vendere più CD invece di quello che ciascuno vorrebbe. Poi qualcuno inventa la tecnologia MP3. All'improvviso la musica può essere scaricata da internet.

L’MP3 è in grado di fare la cosa giusta per il consumatore: gli dà la possibilità di scegliere.

Così se un normale CD costa 10 dollari e contiene due singoli che il consumatore desidera, per lui potrebbe essere più sensato acquistare quei singoli separatamente - insieme ad altri a sua scelta - in MP3 per un dollaro e mezzo l’uno. L’acquirente non è più intrappolato in una situazione dispotica, non deve più sottomettersi alle regole avide delle case discografiche che non sono disposte a dargli più del minimo indispensabile. C’è un ottimo motivo per cui l’industria discografica è spaventata a morte da MP3 e dalle tecnologie associate a questo formato, Napster e Gnutella. Il prezzo dell’acqua è salito tanto che per qualcun altro è diventato redditizio trovare un nuovo modo per farla arrivare da altre parti.

Ma questo è un settore che per tradizione cerca di controllare i consumatori, tramite la musica che decide di pubblicare o tramite i copyright e la tecnologia. È il settore che negli anni Sessanta fece di tutto per impedire ai consumatori di copiare la musica sulle cassette, quando questa tecnologia venne immessa sul mercato. Le case discografiche ritenevano che le cassette fossero il mezzo ideale per permettere alla gente di disobbedire alle leggi sul diritto d’autore, per cui si misero a cercare dei modi per proteggere i propri copyright. Era una pessima scusa. Si nascondevano dietro una morale di facciata e dietro il diritto d’autore quando in realtà stavano semplicemente cercando di mantenere il controllo sulla propria nicchia di mercato. Il fatto è che le cassette non hanno mai danneggiato l’industria discografica. Certo, le persone registravano la musica per il proprio uso personale, ma in realtà questo significava semplicemente che acquistavano più dischi da copiare. Tutto lì. Qualche decennio più tardi, quando uscirono i CD, i lettori venivano fatti in modo che non si potessero registrare perfettamente su cassetta. La paranoia colpisce ancora. Poi vennero le cassette digitali. Avevano una frequenza di campionamento diversa dai CD - 48 kilohertz contro 44,1 -per impedire agli utenti di copiare i propri CD sulle cassette digitali. Ancora una volta il settore discografico cercava di fregare il consumatore per conservare il proprio controllo. Ma nel caso delle cassette digitali, non vi fu mai un vero decollo. Era un po’come cercare di prendere in giro Madre Natura.

Cercando di controllare tutte le nuove tecnologie, l’industria discografica riesce solo a spingere la gente a trovare nuovi modi per aggirare gli ostacoli. Ed è un processo che può continuare all’infinito.

Questo ci porta inevitabilmente ai DVD. Questa volta l’industria dello spettacolo disponeva di un suono e di un’immagine migliori rispetto alle cassette in VHS, oltre a un formato meno ingombrante e a una maggiore facilità d’uso. Ma vi aggiunsero la cifratura per impedire che fossero fatte delle copie. E per sommare il danno alla beffa, aggiunsero anche dei codici di area geografica. I DVD che compravi all’aeroporto di San Francisco non funzionavano in Europa. Per le aziende la cosa aveva una sua logica perversa: Ehi, ragazzi, in Europa possiamo vendere i film a un prezzo più alto. Però dobbiamo essere sicuri che gli europei non possano comprare i film negli Stati Uniti.

L’industria dello spettacolo non era in grado di prevedere l’ovvio? Come al solito, il prezzo dell’acqua sarebbe salito tanto che qualcun altro avrebbe trovato redditizio

inventare un nuovo modo per farla arrivare da altre parti.

Sì, mentre le aziende avide cercavano di controllare la gente tramite la tecnologia, la cifratura del DVD veniva crackata, e nemmeno da persone che volevano copiare i DVD, ma da gente che voleva vederli in Linux. Erano persone che di fatto volevano comprare dei DVD, ma non potevano: i dischi sarebbero stati inutilizzabili sui loro computer. Le mosse che le aziende avevano fatto per proteggersi si rivoltarono contro di loro: impedivano soltanto che il mercato si espandesse e creavano un incentivo a crackare la cifratura dei DVD. Ancora una volta, la strategia a breve termine si è rivelata sbagliata.

L'industria dello spettacolo è solo un esempio. La stessa cosa è successa per anni con il software. È per questo che la strategia di Microsoft del software in *bundle* è destinata a fallire. I prodotti open source, d'altra parte, non possono essere utilizzati in modo dispotico, perché sono *free.* Se qualcuno cercasse di mettere qualcosa in *bundle* con Linux, altri potrebbero semplicemente separare i due prodotti e venderli come li vuole la gente.

Cercare di controllare le persone tramite la tecnologia è doppiamente inutile. È una cosa che finisce sempre per danneggiare l'azienda e per inficiare l'accettazione della tecnologia. Un esempio recente è quello di Java, che ha perso molta dell'attrattiva che aveva inizialmente. Cercando di controllare l'ambiente Java, la Sun Microsystems l'ha fondamentalmente perso. Java va ancora abbastanza bene, ma di certo non ha sfruttato tutto il suo potenziale.

Sun non cercava di guadagnare da Java in sé, ma l'azienda considerava il linguaggio di programmazione come un modo per adattare meglio i computer al proprio utente e liberarsi dalla morsa di Microsoft (nonché naturalmente per vendere più hardware Sun). Ma anche se non stavano cercando di guadagnare direttamente da Java, pensavano di doverlo controllare come fenomeno e come microbo. Tutti i loro contratti di licenza erano appesantiti da questi tentativi di controllo.

Un buon prodotto. Ma il problema era che stavano cercando di fare le scarpe a Microsoft. Erano motivati dalla paura, dal disgusto e dall'odio, che a metà degli anni Novanta sembravano costituire l'approccio più diffuso agli affari. (Pensate a quel testo dei Grateful Dead: *«Ain't no time to hate.»)* E dato che provavano tanto odio e tanta paura nei confronti della Microsoft, presero tutte le decisioni sbagliate in termini di licenza. Resero difficile a tutti, anche ai loro partner, utilizzare il prodotto. È per questo che aziende come la Hewlett Packard e la IBM finirono per decidere di implementare Java autonomamente. «Affanculo la Sun», pensarono.

La Sun provò a standardizzare Java presso due organizzazioni diverse, e ogni volta fallì per colpa di questa faccenda del controllo. Da un lato la Sun voleva standardizzare il linguaggio. Ma allo stesso tempo non voleva perdere il controllo. Così gli enti preposti alla standardizzazione dissero qualcosa del tipo: «Ehi, questa non è roba solo vostra.» Il risultato fu che la Sun dovette alzare i tacchi. È un ottimo esempio di un'azienda che cerca di controllare la tecnologia in modi che non hanno senso per le persone che di fatto utilizzano quella tecnologia. Questo tipo d'atteggiamento si rivela sempre un fallimento per l'azienda, e fa fallire anche la tecnologia (o allunga i tempi necessari perché venga accettata).

Pensate invece alla strategia «If-you-love-something-set-it-free» di Palm

Computing. Quelli della Palm hanno aperto il proprio ambiente di sviluppo e la propria piattaforma non solo alle aziende ma anche agli individui che vogliono scrivere programmi per i loro prodotti. Hanno aperto le loro API *(Application Program Interface,* interfaccia per programmi applicativi) e hanno reso più facile ottenere liberamente i loro strumenti di sviluppo. Il risultato è stata la nascita di un vero e proprio sottosettore industriale intorno al Palm Pilot. La Palm è diventata un fenomeno, non solo una delle tante aziende che cerca di accaparrarsi un nuovo mercato. Così oggi ci sono aziende che vendono giochi che girano su Palm Pilot e programmi di calendario più avanzati di quelli offerti dalla Palm stessa. Oggi i consumatori possono scegliere ciò che vogliono e tutti ne traggono dei benefici, soprattutto la Palm che essendosi aperta a questo modo ha conquistato un mercato più ampio.

La Handspring sta facendo la stessa cosa con la propria macchina, il Visor. È un minicomputer in concorrenza con il Palm, che utilizza il sistema operativo Palm, e l'azienda ha portato l'apertura un passo più avanti consentendo la creazione di periferiche hardware come ricevitori GPS ed estensioni per telefonia mobile. Come la Palm, la Handspring sta creando una comunità di aziende che supportano la nuova piattaforma.

La Sun avrebbe potuto consentire a tutti di creare il proprio Java - senza alcuna complicazione - guadagnando sul fatto che loro potevano fare un lavoro migliore di chiunque altro. È il segno di un'azienda che non è accecata dall'avidità o dalla paura della concorrenza. È il segno di un'azienda che crede in sé stessa. E non ha tempo per odiare.

Il bello deve ancora venire

C’è qualcosa di più irritante al mondo degli oracoli economici? Di quei tipi tronfi che pretendono di sapere dove ci porterà il folle ottovolante della tecnologia? Immagino che anche loro servano a qualcosa. A popolare le tavole rotonde delle conferenze tecnologiche che - identiche in tutto le une alle altre - spuntano come funghi velenosi nel nostro bel praticello fiorito. La gente che spera di fare i soldi con le tendenze tecnologiche spende migliaia di dollari per sentirli parlare in queste tavole rotonde. Grazie a loro un esercito di portieri d’albergo, camerieri e baristi può conservare il proprio posto di lavoro, per cui immagino che a qualcosa servano anche questi personaggi.

E adesso David mi dice che dovrei scrivere anch’io uno di quei capitoli tipo Il futuro del settore. Mi sembra un po’una stupidaggine, ma in fondo David non mi ha lascialo annegare quando ce ne siamo andati insieme a fare boogie-board, per cui se pensa che i lettori troveranno più interessante il futuro del settore informatico rispetto al senso della vita, chiuderò la bocca e mi metterò a scrivere.

Ma...

Vorrei fosse messo a verbale che in vita mia non sono mai stato bravo a fare previsioni su niente, per quanto riesca a ricordare. Avevo previsto che il piccolo sistema operativo che avevo iniziato a scrivere per me stesso un giorno si sarebbe diffuso in tutto il mondo? No. Cavoli, se è stata una sorpresa. La mia unica scusa è che sembra che questo giochetto della sfera di cristallo non riesca troppo bene a nessun altro, e se per me il successo di Linux è stata una sorpresa, per gli altri è stato un vero e proprio fulmine a ciel sereno. Quindi probabilmente sono stato più bravo degli altri, come oracolo. E chi può dirlo? Forse dopo avere scritto questo capitolo diventerò famoso come il Nostradamus dei nostri giorni.

O forse no. In ogni caso, allacciatevi le cinture: si parte.

Naturalmente possiamo guardare alle esperienze del passato. Possiamo analizzare in ogni miserrimo dettaglio i come e i perché del crollo di un’azienda apparentemente invincibile come AT&T, e possiamo predire che basterà avere un po’di pazienza per vedere crescere l’edera anche sugli edifici verdini di Redmond. Come alla stellina televisiva del momento prima o poi spunteranno le prime rughe e si sgonfieranno le bocce siliconate, anche l’eroe del business di oggi verrà soppiantato da un modello nuovo e più ispirato; e l’azienda dell’eroe, per quanto faccia per reinventarsi (o come altro dicono gli esperti di marketing), finirà per ammosciarsi e riempirsi di rughe, come l’AT&T.

Chiamatela evoluzione. Certamente non ci vuole Einstein per capirlo. Nessun business può vivere per sempre. Fortunatamente.

Ma qual è il motore di questa evoluzione? Esiste un’evoluzione tecnologica fondamentale e intrinseca che un giorno porterà i computer ad assumere il controllo lasciandosi alle spalle la razza umana, come qualcuno sembrerebbe credere? O a

generare l'avanzamento tecnologico è solo una sorta di casuale ineluttabilità del progresso, una cosa tipo «Avanti tutta e affanculo i siluri»?

Io credo di no.

La tecnologia è ciò che noi ne facciamo e nessuna azienda, nessuna tecnologia cambierà la natura fondamentale dei bisogni e dei desideri umani. Come per qualsiasi altra cosa, l'evoluzione porterà lentamente ma ineluttabilmente la tecnologia ad allontanarsi dalla semplice sopravvivenza, ci porterà attraverso una società basata sulla comunicazione e infine ci accompagnerà nel regno del divertimento (attenzione, allarme *déjà vu:* sì, avete già visto questa teoria in questo libro e - se resterete con me fino alla fine - la vedrete ancora).

Gli uomini sono destinati a essere animali socievoli, e la tecnologia li seguirà.

Quindi dimenticate tutte le previsioni su ciò che la tecnologia può fare nei prossimi dieci anni. Sono del tutto irrilevanti. Siamo riusciti a mandare un uomo sulla Luna più di trent'anni fa e da allora non ci siamo mai tornati. Io sono convinto che fondamentalmente questo sia successo perché abbiamo scoperto che la Luna è un posto abbastanza palloso, senza un minimo di vita notturna - un po'come San Jose. E quindi l'uomo non ha *voluto* ritornarci e tutta la tecnologia che nel frattempo abbiamo ammassato non significa nulla. La Luna resta disabitata.

Ciò che importa veramente quando si parla del futuro della tecnologia è ciò che la gente *vuole.* Una volta capito questo, la sola domanda che resta è in quanto tempo si potrà avviare una produzione di massa di questa cosa per renderla abbastanza economica da permettere alla gente di averla senza rinunciare a nessuna delle *altre* cose che vuole. Il resto non conta.

Devo fare una piccola digressione. La cosa che fa vendere i prodotti, naturalmente, è la *percezione,* non la realtà. Le navi da crociera vendono la percezione della libertà, di mari aperti, di buon cibo e di romanticismo tipo *Love Boat.* Chi se ne frega se la vostra cabina è microscopica e vi sentite come un uccello in gabbia!

E questo cosa significa? Spiega per esempio perché la gente non fa altro che parlare della Sony PlayStation 2, la periferica tecnologica del momento. (Sto scrivendo pochi giorni dopo che la PS2 è stata messa sul mercato negli Stati Uniti, nell'ottobre 2000.) Una vera e propria incarnazione della società del divertimento!

E tutto ciò ci mostra anche come i personal computer abbiano un problema di percezione. Il settore dei PC ha evidentemente paura delle console per videogiochi, soprattutto perché sono considerate più semplici da utilizzare, più divertenti ed economiche, mentre i PC sono ritenuti perlopiù complicati e costosi. A volte addirittura ostili.

Personalmente la cosa mi convince anche del fatto che se se tra quindici anni saremo ancora lì a parlare tanto di sistemi operativi, vorrà dire che avremo fatto qualche grosso errore. Potrà sembrare strano che lo dica uno che è famoso per avere scritto il proprio sistema operativo, ma il fatto è che - da un punto di vista statistico - *nessuno* vuole un sistema operativo.

In effetti nessuno vuole nemmeno un computer. Ciò che tutti vogliono è un giocattolo magico che possa essere usato per navigare in internet, scrivere le tesi di laurea, giocare, controllare il proprio conto corrente e così via. Il fatto che per fare tutto questo ci sia bisogno di un computer e di un sistema operativo è una cosa a cui

la maggior parte delle gente preferirebbe non dover nemmeno pensare.

È per questo che a molti analisti piace l'idea di una periferica come la Sony PlayStation 2, che fa buona parte del lavoro di un computer senza avere l'aria complicata e spaventosa dei PC. In questo senso l'interfaccia fisica del PC è assolutamente insensata da un punto di vista tecnologico, perché nelle nostre case stanno entrando sempre più computer senza che ci rendiamo conto di quanto siano complessi.

Quindi, se dovessi dire quale sarà la Microsoft del futuro opterei per la Sony, se riuscirà a tenere insieme tutti i propri pezzi. Lo so, non è una di quelle previsioni sconvolgenti alla Nostradamus. Ci sono altre persone che la pensano allo stesso modo, ma io sto cercando di spiegarvi *perch*é sta succedendo una cosa del genere.

Non fraintendetemi: non prevedo la scomparsa del PC, come in passato hanno fatto in molti (sbagliando senza eccezioni). I punti di forza fondamentali dei PC non sono scomparsi: sono il coltellino svizzero dell'informatica. Abbastanza complicati da spaventare le persone a cui non piace la tecnologia, tanto complicati proprio *perch*é non sono fatti su misura per svolgere una sola funzione. È proprio la flessibilità a renderli attraenti.

E poi «un unico anello per governarli tutti, e legarli nell'oscurità»: la comunicazione. Ovunque. Non potete vivere senza controllare l'e-mail almeno ogni mezz'ora? Nessun problema, amici emaildipendenti. Potete concedervi il lusso e il leggero senso di colpa di prendervi una giornata libera, andare in spiaggia e restare comunque in contatto con ciò che succede in ufficio. Ricordate: ciò che vende non è la *realtà* di essere in vacanza, ma la *percezione* della libertà. Diciamocelo: le dimensioni contano, perlomeno per far apparire semplici e non minacciose le meraviglie della tecnologia.

E in tutto questo dove finirà Linux (e più in generale l'open source)? Non ve ne accorgerete nemmeno. Sarà dentro le macchine della Sony. Non lo vedrete mai, non lo saprete mai, ma sarà lì, a far funzionare tutto quanto. Sarà nel telefono cellulare che è al tempo stesso il vostro personalissimo nodo di comunicazione con il resto dei vostri aggeggi elettronici quando siete lontani dalla vostra rete locale *wireless.*

Vedrete. È solo una questione di tempo. E di soldi.

Il senso dell’open source

La IBM, come azienda, ha una tradizione: fregare gli altri. Ha fatto i soldi mettendo in catene i propri clienti e assicurandosi che nessun altro potesse vendere loro qualcosa. Una volta la maggior parte delle aziende informatiche funzionavano così. Certe lo fanno ancora oggi. Poi, quando la IBM sviluppò il personal computer, senza volerlo aprì la propria tecnologia in modo che chiunque la potesse replicare. Quel gesto fu fondamentale per la Rivoluzione del PC, la Rivoluzione di internet, la New Economy e per tutti quegli altri modi à *la page* per indicare i cambiamenti tecnologici che stanno avendo luogo nel mondo.

Si tratta dell’esempio migliore per illustrare gli illimitati benefici che possono derivare dalla filosofia open source. Per quanto il PC non sia stato sviluppato usando un modello open source, è un esempio di una tecnologia che è stata aperta in modo che qualsiasi persona o azienda potesse clonarla, migliorarla e venderla. Nella sua forma più pura, il modello open source consente a tutti di partecipare allo sviluppo di un progetto o al suo sfruttamento commerciale. Linux è ovviamente l’esempio di maggior successo. Partito nella mia stanzetta incasinata di Helsinki, è cresciuto fino a diventare il più grande progetto collaborativo nella storia del mondo. È iniziato come un’ideologia comune tra alcuni sviluppatori di software che ritenevano che il codice sorgente dovesse essere condiviso liberamente. Lo strumento più potente del movimento era la General Public License, l’anticopyright. Questa ideologia ebbe un’evoluzione e diventò un metodo per lo sviluppo continuativo della tecnologia migliore. E poi crebbe ancora fino a essere adottata un po’ovunque sul mercato, come dimostrano la diffusissima adozione di Linux come sistema operativo dei server web e le sue IPO incredibilmente generose.

Ispirato dall’ideologia, l’open source ha dimostrato il proprio valore tecnologico e la propria adattabilità al mercato. Oggi l’open source si sta espandendo ben oltre gli ambiti tecnologici ed economici. Alla facoltà di giurisprudenza della Harvard University, i professori Larry Lessig (che ora è a Stanford) e Charles Nesson hanno applicato il modello open source alla legge. Hanno avviato l’Open Source Project: avvocati e studenti di giurisprudenza pubblicano opinioni e ricerche sul sito internet del progetto per contribuire allo sviluppo di argomentazioni e memorandum contro il Copyright Extension Act statunitense. L’idea è che le argomentazioni più forti verranno sviluppate a mano a mano che il maggior numero di menti lavoreranno al progetto e una montagna di informazioni verranno generate tramite i contributi e le risposte a questi. Nel sito viene riassunto alla perfezione il vantaggio rispetto all’approccio tradizionale: «Ciò che perdiamo in segretezza, ci aspettiamo di guadagnarlo in profondità delle fonti e vastità delle argomentazioni». (Per passare a un altro contesto: con un milione di occhi a disposizione, tutti i bachi del software scompariranno.)

Si tratta di un espediente basato sul modo in cui è stata condotta per anni la ricerca

accademica, ma è un espediente sensato su più fronti. Pensate per esempio a come questo approccio potrebbe rendere più veloce lo sviluppo delle cure per determinate malattie. Oppure a come, con le migliori menti al lavoro insieme, si potrebbe rafforzare la diplomazia internazionale. Il mondo diventa sempre più piccolo, il ritmo della vita e degli affari sempre più veloce, le tecnologie e le informazioni sempre più diffuse, e le persone iniziano a comprendere che un approccio individualista ed egocentrico è sempre meno adatto a questa realtà.

La teoria alla base dell'open source è semplice. Nel caso di un sistema operativo, il codice sorgente - le istruzioni di programmazione sottese al sistema - è aperto e gratuito. Tutti possono migliorarlo, cambiarlo, sfruttarlo. Ma questi miglioramenti, questi cambiamenti e questi utilizzi devono essere messi gratuitamente a disposizione di tutti. Pensate Zen. Il progetto non appartiene a nessuno. E di tutti. Quando si apre un progetto, si assiste a un progresso rapido e continuo. Con squadre di collaboratori attivi in parallelo, i risultati possono arrivare più in fretta e con maggior successo rispetto a una situazione in cui i lavori si svolgono dietro porte chiuse.

È questo che abbiano sperimentato con Linux. Provate a immaginare: invece di un team di sviluppo chiuso, avete dalla vostra parte un mostro. Potenzialmente, milioni di menti estremamente brillanti contribuiscono a un progetto, e siete supportati da un processo tra pari che non ha... ehm... pari.

La prima volta che qualcuno sente parlare dell'approccio open source, lo trova assurdo. É per questo che ci sono voluti anni perché si affermasse il suo messaggio. Non è stata l'ideologia a dare il successo al modello open source. Ha iniziato a conquistare l'attenzione della gente quando è risultato evidente che l'*open source* era il metodo migliore per sviluppare e migliorare una tecnologia di altissima qualità. E ora sta conquistando il mercato, un risultato che ha contribuito ulteriormente all'accettazione diffusa dell'open source. É stato possibile creare delle aziende intorno a diversi successi ad alto valore aggiunto, l'open source è stato usato come un modo per rendere popolare la tecnologia. Quando entrano in gioco i giochi, è più facile convincere la gente.

Uno dei pezzi più incompresi del puzzle open source è il perché così tanti ottimi programmatori accettino di lavorare assolutamente gratis. È necessario dire qualcosa sulle motivazioni. In una società in cui la sopravvivenza è più o meno assicurata, il denaro non è la motivazione più forte. É stato provato senza ombra di dubbio che le persone lavorano al meglio quando sono guidate dalla passione. Quando si divertono. È vero per gli sceneggiatori, per gli scultori, per gli imprenditori e - naturalmente - per i programmatori informatici. Il modello open source dà alla gente la possibilità di vivere della propria passione. Di divertirsi. E di lavorare con i migliori programmatori del mondo, invece che con i quattro gatti che sono stati assunti dalla loro azienda. Gli sviluppatori open source si danno da fare per guadagnarsi la stima dei propri colleghi. E questa sì che è una motivazione forte.

Sembra che Bill Gates non riesca ad afferrare questo concetto. È possibile che oggi si senta in imbarazzo ripensando a una domanda retorica che pose nel 1976? «Una delle cose che fate è impedire che venga scritto del buon software. Chi potrebbe permettersi di fare un lavoro professionale gratis?», scrisse Bill in una lettera ai programmatori open source.

In effetti un modo per comprendere il fenomeno open source consiste nel pensare a come la scienza veniva percepita dalla religione parecchi secoli fa (e, da parte di alcune persone, ancora oggi). La scienza era vista in origine come qualcosa di sovversivo e pericoloso per l'establishment: fondamentalmente la stessa cosa che a volte le software house pensano dell'open source. É proprio come la scienza non era nata da un tentativo di minare l'establishment religioso, l'open source non è stato concepito per far esplodere quello informatico. É nato per produrre la migliore tecnologia e per vedere dove va a finire.

La scienza in sé non permette di fare soldi. Sono stati gli effetti secondari della scienza a creare tutta quella ricchezza. Lo stesso vale per l'open source. Consente di creare settori secondari che sfidano le aziende esistenti, un po'come i derivati della scienza sfidavano la chiesa. Potete trovare delle piccole aziende come VA Linux che sfruttano l'open source e all'improvviso riescono a competere con le società tradizionali. Come disse Sir Isaac Newton, si ergono sulle spalle dei giganti.

A mano a mano che l'open source prende l'abbrivio nell'economia mondiale e i suoi sviluppatori sono sempre più riconosciuti, il loro valore sul mercato del lavoro continua a crescere. Le aziende spulciano le credit list tradizionalmente allegate ai contributi dei software open source. E poi danno istruzioni ai loro uffici per le risorse umane di consegnare una carriola di soldi e stock option nelle mani dei potenziali dipendenti. Poco fa ho detto che il denaro non è il fattore motivazionale più importante, e non ho cambiato idea nel giro di poche righe. Ma non è male ricevere un bel gruzzolo in cambio del proprio duro lavoro. E certamente torna comodo quando devo riempire il serbatoio della mia BMW.

Come quelli della scienza, gli effetti secondari dell'open source sono infiniti. L'open source sta creando cose che fino a poco tempo fa erano ritenute impossibili, e sta aprendo mercati nuovi e inesplorati. Con Linux, come con altri progetti open source, le aziende possono creare le proprie versioni, apportare i propri cambiamenti, cosa assolutamente impossibile altrimenti. È incredibile vedere come quasi tutto ciò che è stato fatto con Linux non era minimamente stato previsto quando partimmo. Sta decollando anche in Cina. In passato lo sviluppo di software in Asia consisteva soprattutto nella traduzione di programmi americani o europei. Oggi c'è gente da quelle parti che usa Linux per sviluppare il proprio software. E io sono molto fiero del tizio che è venuto da me al Comdex per mostrarmi una pompa da benzina che girava in Linux. Era un prototipo di pompa da benzina che montava Linux con dei browser per internet in modo che i clienti mentre facevano benzina potessero andare a vedere il sito della CNN. In piedi sulle spalle dei giganti.

È magnifico vedere come la gente possa usare tecnologie come Linux per creare una pompa da benzina migliore. È un tipo di innovazione che probabilmente non sarebbe nata entro i confini di un'azienda, perché se tu fossi un'azienda che vuole portare Linux sul mercato sceglieresti la via più ovvia, che al momento è il mercato dei server o quello dei desktop di fascia alta. Ma l'open source in generale consente alle aziende di prendere le proprie decisioni su ciò che vogliono fare. E così Linux finisce anche nelle periferiche incapsulate, Tivo gira in Linux, Web Slate della Transmeta gira in Linux e Telephony gira in Linux. È così che con l'open source si stanno facendo miliardi di dollari.

È come lasciare che l'universo si prenda cura di sé stesso. Se non controlli la tecnologia, non ne limiti gli utilizzi. La metti a disposizione e le persone prendono le proprie decisioni localmente, la usano come trampolino di lancio per i loro prodotti e i loro servizi. E per quanto la maggior parte di queste decisioni non abbiano molto senso nello schema generale delle cose, funzionano davvero bene. Non è questione di diffondere Linux. È questione di mettere Linux a disposizione e poi lasciare che si diffonda da solo. E questo non vale solo per Linux. Vale per tutti i progetti in open source.

L'open source ha un senso.

Le persone non sono disposte a scherzare con il diritto di parola. È una libertà che hanno difeso con la vita. La libertà è sempre una cosa che devi difendere con la vita. Ma inizialmente non è una scelta facile da fare. E lo stesso vale per l'apertura. Devi solo decidere di aprirti. È una presa di posizione difficile da assumere all'inizio, ma alla fine crea una maggiore stabilità.

Pensate alla politica. Se la logica che viene usata contro l'open source venisse applicata al governo, avremmo sempre un sistema a partito unico. È evidente che un sistema a partito unico è decisamente più semplice del nostro sistema pluripartitico, il sistema politico aperto in cui opera la maggior parte del mondo. Con un solo partito non ti dovresti preoccupare di trovare accordi con altre persone. Il ragionamento seguente sarebbe questo: il governo è una cosa troppo importante per affidarlo ai tira-e-molla dell'apertura. Per un qualche motivo la gente nota subito la fallacia di questa argomentazione quando riguarda la politica e il governo, e non quando concerne gli affari. Anzi: nel mondo degli affari l'idea di apertura fa innervosire le persone.

Le argomentazioni che vengono utilizzate dalle aziende che rifiutano di aprire le proprie tecnologie sono convincenti: *Non* è *cos*ì *che si fanno le cose,* dicono i dirigenti. È spaventoso. La gente ha paura del cambiamento, in parte perché non sa cosa può aspettarsi. Restando avvinghiata allo status quo, un'azienda può avere un'idea più precisa di dove arriverà, e a volte questa cosa sembra più importante anche rispetto al successo. Vi sono aziende che preferiscono avere successo in modo prevedibile che avere un *enorme* successo in modo imprevedibile.

Non è facile per un'azienda trasformare un prodotto esistente in un prodotto open source. C'è tutta una serie di questioni spinose. In primo luogo nel corso dei mesi e degli anni passati a sviluppare il prodotto, l'azienda ha creato una grossa mole di conoscenze interne. Questa proprietà intellettuale interna è il pane quotidiano dell'azienda. L'organizzazione non desidera abbandonare la proprietà intellettuale che la tiene in vita. Ma l'esistenza stessa di questa conoscenza interna crea delle barriere per chi sta all'esterno. Disincentiva la sua partecipazione.

Io però ho visto delle aziende trasformarsi da chiuse ad aperte. Per esempio la Wapit, una società finlandese che crea servizi e infrastrutture di supporto per diverse periferiche interattive. Questo progetto coinvolgeva il server web tipo telefono a muro creato dall'azienda. Per loro, la decisione di mettere in open source i propri software è perfettamente sensata. Vogliono costruire il proprio servizio, ma prima devono costruire l'infrastruttura. Per farlo è necessario scrivere moltissimo software. É un male inevitabile. Così invece di vederla come una decisione di mettere le loro proprietà intellettuali a disposizione degli altri, alla Wapit la pensano in questo modo:

la scrittura del software richiede molto tempo di programmazione, ma non crea necessariamente valore per il solo fatto di essere interna all'azienda.

C'erano alcuni elementi che lavoravano a favore della Wapit. Primo: non era un grosso progetto. Secondo: la decisione di passare all'open source venne presa in una fase iniziale dell'esistenza dell'azienda. I dirigenti ritenevano di avere abbastanza risorse interne per sviluppare il progetto, ma volevano ottenere qualcosa di più di quanto avrebbero potuto produrre internamente. Decisero anche che l'open source era un buon modo per spingere il wap come standard su cui altri potessero lavorare.

All'inizio della partita l'azienda mi chiese dei consigli e io dissi loro che dovevano combattere la tentazione di prendere le decisioni internamente. Consigliai loro che - nel caso fosse necessario tenere delle riunioni in cui prendere delle decisioni - queste riunioni non fossero chiuse agli esterni. Riservando all'azienda il processo decisionale, avrebbero corso il rischio di alienarsi gli esterni, che avrebbero fatto fatica a entrare nel cuore del progetto. Questo è uno dei problemi principali quando si crea e si gestisce un progetto open source in ambiente aziendale. É facile parlare di open source, ma la cosa può facilmente degenerare in una specie di incontro di boxe: Noi contro Loro. Molte decisioni vengono prese nel modo più facile: stando seduti a un tavolino della mensa a discutere le diverse possibilità e a sviluppare il consenso senza mai aprire la discussione all'esterno. Chi viene dall'esterno - per quanto le sue opinioni possano essere valide - è tagliato fuori dal semplice fatto che la decisione è già stata presa nella mensa aziendale.

Questo era uno dei problemi che affliggevano la Netscape nei mesi successivi alla sbandierata decisione presa nella primavera del 1998 di rilasciare i codici sorgente (con il nome di Mozilla) del proprio browser di nuova generazione. Ci volle molto tempo prima che il progetto fosse effettivamente degno delle proprie premesse open source. Esisteva uno schieramento di interni della Netscape che non accettavano nemmeno una patch dall'esterno. Tutti là dentro si conoscevano, e se non erano fisicamente seduti attorno a un tavolo della mensa a prendere le decisioni, e era una mensa virtuale in cui si potevano trovare gli interni. Invece di essere vista come un'azienda alle prese con il primo grande esperimento della trasmigrazione in open source di un progetto commerciale preesistente, la Netscape fu trattata molto male dalla stampa. Quando si sparse la voce delle sue perdite, l'azienda non potè più accampare grandi scuse morali. E questo l'aiutò ad aprirsi all'esterno. Oggi il progetto sembra essere molto più dinamico.

Quando sente parlare per la prima volta della possibilità di aprire un progetto commerciale esistente, la gente tende a pormi sempre le stesse domande. Un esempio: come si sentirebbero le persone interne all'azienda rispetto alla possibilità di avere degli esterni che producano un lavoro migliore del loro, e per di più davanti agli occhi di tutti? Secondo me dovrebbero sentirsi perfettamente a loro agio rispetto a quest'idea e rispetto al fatto di essere pagati per non fare la maggior parte del lavoro. Da questo punto di vista l'open source - o qualsiasi cosa sia aperta, se è per questo - non perdona. È sempre evidente chi riesce a fare un lavoro e chi lo fa nel modo migliore. Non ti puoi nascondere dietro i dirigenti.

L'open source è il modo migliore di sfruttare i talenti esterni. Ma hai comunque bisogno di avere qualcuno all'interno che tenga d'occhio le esigenze dell'azienda.

Questa persona potrebbe anche non essere il responsabile del progetto. In effetti per l'azienda sarebbe un vantaggio se qualcuno all'esterno si assumesse volontariamente la responsabilità del progetto. Se un esterno fa un lavoro migliore, è meglio così. Il problema è che questo esterno potrebbe portare il progetto in una direzione che non soddisfa le necessità dell'azienda. E così l'azienda deve badare ai propri bisogni. L'apertura di un progetto potrebbe mettere in grado l'organizzazione di ridurre le proprie risorse locali, ma questo non vuol dire che se ne possa disfare. Il progetto si potrebbe espandere molto più di quanto non sarebbe stato possibile all'interno di una singola azienda. Le risorse esterne consentono di costruire un sistema più economico, completo ed equilibrato, ma c'è un rovescio della medaglia: il sistema espanso non tiene più conto solo dei bisogni dell'azienda. Potrebbe prendere in maggior considerazione quelli dei clienti.

Probabilmente il problema principale dell'intero processo consiste nel rinunciare al controllo e accettare il fatto che degli esterni potrebbero saperla più lunga di noi. L'altra difficoltà sta nel trovare un leader tecnologico forte all'interno dell'azienda. Deve essere una persona di cui tutti si fidano su due livelli: quello tecnologico e quello politico. Deve essere una persona in grado di riconoscere il fatto che il progetto potrebbe essere stato sbagliato sin dall'inizio. Invece di cercare di nascondersi rispetto a questi problemi, il leader deve saper convincere tutti che la cosa migliore da fare è tornare indietro e ricominciare da capo, il che significa buttare via parte del lavoro fatto. Non è una cosa che alle persone faccia piacere sentire. Ma, se il messaggio arriva da qualcuno che rispettano, lo accetteranno.

Data la natura delle politiche aziendali e il modo in cui lavorano generalmente le aziende, il leader tecnologico dovrebbe essere una persona con una personalità abbastanza forte. Dovrebbe piacergli lavorare via e-mail ed evitare di schierarsi con gli uni o gli altri. Non userei il termine diplomatico *liason,* perché vorrebbe dire dare per scontato che vi siano due fazioni contrapposte: gli interni e gli esterni. E le cose non dovrebbero andare così. Il leader tecnologico viene pagato dall'azienda per fare open source. Sa - e lo sanno anche tutti gli altri - di non essere pagato per essere d'accordo con le proprie controparti aziendali, ma solo per portare avanti il progetto. Un leader troppo legato all'azienda può essere un problema. La gente si potrebbe fidare delle sue capacità tecniche ma non del suo giudizio politico.

C'è un diplomatico in sala?

È una cosa tipo «Trovatemi un uomo onesto».

È per questo che nel corso degli anni ho sempre cercato di evitare di farmi coinvolgere dalle aziende Linux. E adesso che iniziano a vedersi i soldi la situazione si fa ancora più critica. Con tutti questi dollari che svolazzano in giro, la gente inizia a porsi delle domande sulle tue motivazioni. Per me essere conosciuto come una persona neutrale è molto utile. Non avete idea di quanto sia importante per me mantenere questa neutralità. È una cosa che mi fa impazzire.

Ok. Avete ragione. Dovrei smetterla con le prediche. L'open source non è per tutti, per qualsiasi progetto e per qualsiasi azienda. Ma a mano a mano che la gente inizia a metabolizzare il successo di Linux, comincia anche a capire che non si tratta di una mattana ideata da un gruppetto di liceali brufolosi e idealisti.

Aprite tutto ciò che si può aprire e ne nasceranno nuove possibilità. Io parlo di

open source da quando i giornalisti hanno iniziato a farmi domande in proposito, fondamentalmente negli ultimi cinque anni. All'inizio dovevo continuare a spiegare cosa c'era di tanto bello nell'open source. E sinceramente sembrava di percorrere una strada senza fine. Era come avanzare nel fango.

Ora la gente lo ha capito.

Fama e ricchezza

«Che mi dici del peso della fama?» Ogni tanto me lo chiedono. E lasciate che vi dica che questo «peso non è affatto tale. È divertente essere famosi e le persone famose che dicono il contrario stanno solo cercando di apparire simpatiche e di far pensare a quelli che non sono famosi che stanno meglio loro. La gente si aspetta che tu sia molto umile a proposito della fama e che ti lamenti di come ti distrugge la vita.

Ma diciamocelo in faccia: tutti sognano fama e ricchezza. Io di sicuro lo facevo. Da ragazzino volevo diventare uno scienziato famoso. Tipo Albert Einstein, ma meglio. E così per tutti, no? Se non è uno scienziato è un pilota di formula uno, una rockstar, Madre Teresa o il Presidente degli Stati Uniti d'America.

E a dire la verità arrivarci non è stato affatto brutto. Certo, forse non sono Albert Einstein, ma mi piace l'idea di avere cambiato qualcosa, di avere fatto qualcosa di significativo. E ricevere un riconoscimento per questo rende il tutto ancora più bello. Quindi la prossima volta che sentirete qualcuno lamentarsi perché è ricco e famoso, ignoratelo. Lo sta facendo solo perché ci si aspetta che dica qualcosa del genere.

Quindi sono tutte rose e fiori? Certo che no. Ci sono sicuramente degli svantaggi nell'essere conosciuti. Non è che la gente mi fermi per strada (o perlomeno succede raramente), ma tra le montagne di e-mail che ricevo ci sono anche dei messaggi a cui è davvero difficile rispondere, ma è difficile anche ignorarli. Cosa puoi dire a una persona che ti chiede di scrivere un elogio funebre per suo padre, che tu non hai mai conosciuto? Non ho mai risposto a quel messaggio, e ancora oggi mi sento in colpa per questo. Per qualcuno era una cosa molto importante, per me soltanto un fastidio.

E che cosa puoi dire a una persona che ti chiede di tenere il discorso d'apertura di una conferenza a cui non hai il tempo né la voglia di partecipare? Come fai a far capire alla gente che hai smesso da molto tempo di ascoltare i messaggi in segreteria telefonica senza sembrare uno stronzo? Cosa che peraltro, in fondo, effettivamente sei. Non è che mi interessi sinceramente di qualunque cosa solo perché mi interesso con tutto il cuore a ciò che mi ha dato la fama, Linux.

Naturalmente alla fine diventa molto facile dire di no. O addirittura ignorare le richieste; uno dei motivi per cui adoro l'e-mail è che è comodissima ed estremamente facile da ignorare: cos'è un messaggio in più tra le centinaia che ricevo ogni giorno? Il mezzo di comunicazione è talmente separato dalla persona che solo raramente diventa abbastanza personale da farti sentire in colpa se lo ignori. Succede (vedi sopra), ma non molto spesso. E anche quando non lo ignori, dire «no» per e-mail è molto più facile che farlo di persona o al telefono.

Il problema è fondamentalmente ciò che la gente si aspetta dalle persone famose. Non è assolutamente possibile essere all'altezza di quelle aspettative, ma non puoi fare a meno di pensare che dovresti almeno provare a esserlo. In parte è questo che ha reso tanto difficile scrivere questo libro: cercare di scrivere qualcosa di ragionevolmente personale e al tempo stesso non deludere chi si aspettava qualcosa

di diverso.

E buona parte di queste aspettative sono decisamente stupide. Ho spesso la sensazione che la gente si aspetti che io sia una specie di monaco moderno, che conduce una vita frugale e solitaria. Tutto perché ho pensato che rendere Linux aperto e metterlo a disposizione gratuitamente su internet fosse una buona idea, e perché non ho scelto un approccio commerciale al software. Così io ora mi sento un po'in colpa e mi metto sulla difensiva rispetto al fatto che in realtà mi piaccia spendere il denaro e che finalmente sono passato dalla mia vecchia Pontiac Grand Am a un'auto più divertente[12].

E questo porta alla ribalta la seconda domanda dopo quella sul «peso della fama». Ovvero: «Il successo rovinerà Linus (e/o Linux)?» Mi trasformerò in un marmocchio viziato ed egocentrico che scrive autobiografie perché gli piace l'idea di vedere il proprio nome stampato e per pagare le rate della sua auto nuova?

La risposta, naturalmente, è sì.

Prendete una persona la cui filosofia è sempre stata quella di divertirsi e di fare qualcosa d'interessante, dategli soldi e notorietà e cosa vi aspettate che accada? Che diventi all'improvviso un filantropo? Temo proprio di no. L'idea di dare del denaro in beneficenza non mi è mai nemmeno passata per la testa prima che David mi facesse una domanda in proposito, mentre lavoravamo a questo libro. L'ho guardato senza dire una parola. L'unica cosa che mi è venuta in mente è stata «Shave the whales»[13]. Non sono nato per avere grandi responsabilità finanziarie.

Il successo cambia il nostro modo di fare le cose? Sì. Linux era molto diverso quando aveva solo cinquanta utenti,tutti informatici, rispetto ai 25 milioni (o che so io) di persone qualunque che oggi lo utilizzano almeno occasionalmente. E Linux era molto diverso quando le sole persone che ci lavoravano lo facevano solo perché lo trovavano divertente e interessante, senza gli interessi commerciali che oggi sono tanto evidenti.

E lo stesso vale per l'essere umano Linus. Le cose cambiano, e dire che non è vero non cambia la realtà dei fatti. Linux non è lo stesso movimento di cinque anni fa e Linus non è la stessa persona di allora. E parte di ciò che ha reso Linux tanto interessante per me è proprio il fatto che non è rimasto lo stesso, che ne sono emersi continuamente nuovi aspetti. Con il successo non cambiano solo gli aspetti tecnologici, ma anche questioni che riguardano il significato intrinseco di Linux. Se non fosse così, la vita sarebbe troppo noiosa.

Dunque anziché usare la parola «rovinare», preferirei dire che il successo commerciale ha «cambiato sia Linux sia me. Non arriverei a dire che ci ha fatti «crescere» - credo che per quanto mi riguarda avere tre figli è stato un evento

[12] Non c'è niente di male nella Pontiac Grand Am, è una buona macchina. Probabilmente è anche la macchina più diffusa in tutti gli Stati Uniti e alcuni giornalisti hanno trovato interessante che io avessi un'auto tanto mostruosamente normale. Cavoli, non è nemmeno giapponese! La gente inizia a rispettarmi un po'di meno quando ammetto di avere passato delle ore a scervellarmi per decidere il colore della mia nuova auto, una meno prosaica BMW Z3. Ricordate: «Solo per divertirsi». Quell'auto è assolutamente inutile per qualsiasi cosa non sia il puro divertimento. Ed è così che mi piace.

[13] Gioco di parole intraducibile. Si tratta della deformazione delmotto animalista «Save the whales» (Salvate le balene). «To shave» significa radere, fare la barba, per cui l'espressione si potrebbe rendere in italiano come «Depilate le balene:». [n.d.t]

decisamente più critico - ma solo che ci ha «cambiati;». Ci ha cambiati in meglio, per molti versi, ma ci ha anche resi meno puri. Linux una volta era una cosa *solo* per informatici, un rifugio per *geek*. Un bastione della purezza, in cui era quasi solo la tecnologia a contare.

Oggi non è più così. Linux ha ancora un forte background tecnico, ma il fatto di avere milioni di utenti rende tutti consapevoli del fatto che dobbiamo stare molto più attenti a ciò che si fa. La compatibilità con le versioni precedenti è diventata all'improvviso un elemento importante, e un giorno, tra vent'anni, arriverà qualcuno a dirci che ne ha avuto abbastanza e creerà un suo sistema operativo chiamato «Fredix»[14]. Senza tutta la zavorra storica che Linux avrà accumulato. Ed è esattamente così che deve andare.

Ma ciò che mi fa sentire tremendamente fiero è il pensiero che anche quando arriverà Fredix, le cose non saranno più le stesse. Se non altro Linux ha fatto vedere alla gente un nuovo modo di fare le cose, ha mostrato loro come l'open source possa consentire alle persone di lavorare a partire dal lavoro di altri. L'open source era in circolazione da un pezzo, ma Linux l'ha portato all'attenzione di tutti. Così quando arriverà Fredix, non dovrà cominciare da zero. E il mondo sarà diventato un posto un po'migliore.

---

[14] Oppure «Diannix». Spero che tra vent'anni l'informatica avrà superato la fase dominata dai maschi in cui attualmente ci troviamo...

Quasi un anno dopo che avevamo iniziato a lavorare a questo libro, io e Linus passammo un venerdì sera alla pista di auto-scontri dove avevamo gareggiato qualche mese prima. Questa volta Linus mi batté alla grande: fu più veloce e più bravo negli scontri. Più tardi, mentre mangiavamo cibo turco, io diedi la colpa della mia mediocre prestazione a una pessima giornata lavorativa.

Lui mi guardò e disse: «Be', dovrai restarci ancora per tre mesi.»

«Perché?»

«Non è tra tre mesi che scade il tuo primo blocco di stock option?»

Sto riportando questo dialogo perché la sera della nostra prima competizione alla pista automobilistica Linus mi aveva confessato che la sua memoria era tanto pessima che doveva sempre chiedere a Tove di ricordargli il proprio numero di telefono. All'improvviso ora si ricorda la data di scadenza delle stock option di un'altra persona, e può anche dirmi dove eravamo quando glielo avevo detto. Un anno prima sembrava perfettamente a suo agio nel ruolo del professore con la testa tra le nuvole, incapace di ricordare i particolari di qualcosa di meno significativo della Teoria delle Superstringhe o della capacità di memoria dei suoi primi computer. Ora sembra estremamente presente.

In gennaio ce ne stavamo seduti nella mia vecchia vasca riscaldata e io avevo fatto una battuta sul fatto che il museo storico della marina mi aveva chiesto di donarla alla loro fondazione. In agosto Linus mi fa: «Ehi, quand'è che la porti al museo quella vecchia vasca?» Non deve consultare un'agenda elettronica per ricordarsi quando verrà a trovarlo Avuton. Tiene a mente i dettagli personali di amici e colleghi come non gli accadeva un anno fa. In effetti sa anche cosa fanno i miei amici e colleghi. E per uno le cui prime parole in proposito sono state «In realtà non ricordo molto della mia infanzia», all'improvviso sembra gli siano tornate alla memoria un sacco di cose: «Te l'ho già raccontato che ero così imbarazzato quando mia madre voleva che chiedessi a mio nonno di darmi i 100 marchi finlandesi che mi mancavano per comprarmi il mio primo orologio?».

E questo era solo uno dei modi in cui Linus sembrava essere cambiato nel corso di questo importante anno della sua vita. C'erano altri particolari. In novembre facemmo il viaggio di famiglia a Los Angeles che fa da sfondo alla prefazione di questo libro, «Il senso della vita», in parte perché i Torvalds erano stati invitati alla residenza di Brentwood del consolato finlandese. Prima di partire Linus guardava con occhi sbarrati la corsia dei vini di un supermercato di Santa Clara. «Aiutami a scegliere una bottiglia da regalare», mi disse. «Io non capisco niente di vini.» Dieci mesi più tardi sa qual è la migliore tra due bottiglie di cabernet che si trovano nel minibar del Bodega Bay Lodge. La beviamo guardando un film d'azione in pay-per-view. Lo guardo con la coda dell'occhio: sta lasciando respirare il vino prima di berlo.

E poi c'è la faccenda dell'esercizio fisico. La prima volta che andai a casa di Linus mi sembrò avere il tipico approccio da geek al proprio corpo e al benessere

fisico, una filosofia del tipo «il-mio-corpo-esiste-solo-per-portare-in-giro-la-mia-supermente». Linus sembrava addirittura fiero di non avere mai fatto sport. Tove naturalmente non era d'accordo. I suoi trofei di karate erano tutti esposti in uno scaffale della libreria e appoggiati sopra al televisore c'erano i suoi video di aerobica. Era un argomento di discussione tra loro. «Magari tra cinque anni un dottore mi dirà che devo perdere peso o roba del genere», diceva Linus all'epoca.

A me piace l'esercizio fisico e pensavo che dovessero essere una componente fondamentale delle nostre uscite. Volevo fargli scoprire il surf, ma mi sembrava meglio iniziare con il boogie-board. Un pomeriggio, ai primi di maggio, andammo a Half Moon Bay, affittammo le mute e le tavole e Linus iniziò a protestare al solo pensiero di tuffarsi nelle acque gelide del Pacifico, nonostante la muta a tenuta stagna. Ma nel giro di pochi minuti accadde qualcosa di stupefacente: si divertiva a fare lo slalom tra le onde. «È fantastico!», strillava come un bambino di cinque anni, mentre mi dava il cinque con la mano. Naturalmente dopo un quarto d'ora gli venne un brutto crampo alla gamba - «Sono troppo giù di forma», si giustificò lui - e dovette fermarsi. (Quando gli venne quel crampo Linus si sedette nell'acqua bianca, sembrava non riuscisse ad alzarsi e le onde gli si infrangevano addosso. Il mio primo pensiero fu: «Oh cazzo, se uccido quest'uomo mi ritroverò milioni di nerd alle calcagna. »)

Linus si divertì un sacco a fare tutte le cose in cui ci impegnammo nella fase preparatoria di questo libro: giocare a tennis, sfidarmi in gare di nuoto, salire sulle spaventose attrazioni del parco Great America, giocare a golf. A un certo punto fu evidente che gli interessava più divertirsi a questo modo che stare seduto a parlare nel mio registratore. I bagni nel fango, la risalita a piedi del Tamalpais, il tiro al piattello, qualsiasi cosa. «Dovrei farlo più spesso», mi disse tutto sudato dopo che avevamo giocato a tennis dalle parti di casa mia. Quella volta aveva preso in prestito sia la racchetta sia le scarpe da tennis. Da allora tiene sempre le sue nuove scarpe da ginnastica nel baule dell'auto. Non si sa mai.

Il senso della vita - SECONDA PARTE

Vi è mai capitato di sdraiarvi a terra in una tiepida notte d'estate, guardare le stelle e chiedervi perché siete qui? Qual è il vostro posto nel mondo e cosa dovreste fare della vostra vita?

Be', nemmeno a me.

Ma questo non mi impedisce di avere una mia teoria sulla Vita, sull'Universo e su Tutto (almeno per quanto riguarda la questione Vita, ho una certa esperienza personale). Questa teoria ve l'ho raccontata nella prefazione. E dato che siete arrivati fino a qui, potrei anche spiegarvela un po'meglio.

La mia teoria non mi è saltata alla mente mentre guardavo le stelle, immerso nella meraviglia dell'incommensurabilità di una notte illuminata dalla luna. L'ho inventata mentre preparavo un discorso. Quando si diventa famosi per un qualche motivo, la gente pensa che possiate avere delle idee intelligenti anche su tutti i settori della conoscenza che hanno messo in scacco l'umanità per milioni di anni. E vuole che comunichiate queste idee davanti a una platea di perfetti estranei.

No, non ha alcun senso. Io ho iniziato a lavorare a Linux perché ero un *geek,* non perché ero bravo nelle relazioni pubbliche (per non parlare delle speculazioni filosofiche). Ma non sono molte le cose sensate nella vita, per cui non mi lamento.

Di cosa stavamo parlando? Ah, sì.

Quella volta ero stato invitato a un evento che si teneva a Berkeley e si chiamava «Webrush». Normalmente non l'avrei nemmeno preso in considerazione, ma l'invito era arrivato dal consolato finlandese negli Stati Uniti ed essendo io una persona patriottica (o quantomeno uno che si sente in colpa per il fatto di odiare la neve e di essersi trasferito all'estero) fui tanto stupido da dire: «Ok. *Jag gör det.*»[15]

Ora: nessuno si aspettava che parlassi del senso della vita, e io meno di tutti. Ma quella conferenza riguardava la vita in una società in rete e io ero lì nei panni di quello che si intendeva di internet e come rappresentante della Finlandia. La Finlandia, grazie alla Nokia (la più grande, la migliore e la più bella azienda del mondo, come potrà dirvi qualsiasi finlandese), è molto presa dalla questione comunicazioni ed è proprio una «società in rete». Abbiamo già discusso di come in Finlandia ci siano più cellulari che persone, e delle ricerche in corso per trovare dei modi per impiantare chirurgicamente questi affari nei neonati.

Così me ne stavo lì seduto a chiedermi di quale aspetto della comunicazione avrei dovuto parlare. Ah, dimenticavo di dirvi che la maggior parte degli altri partecipanti alla tavola rotonda sarebbero stati filosofi che parlavano di tecnologie. In fondo eravamo sempre a Berkeley. Ci sono due cose che vengono prese molto seriamente a Berkeley: la politica di Berkeley e i filosofi di Berkeley.

Che cavolo, mi dissi. Se chiamano dei filosofi a parlare di tecnologie, perché un

---

[15] «Ok. Lo faccio»

tecnico non dovrebbe parlare di filosofia? Nessuno mi può accusare di non avere le palle. Mi possono dare del deficiente (e forse in effetti lo fanno), ma non certo del *coniglio.*

Non a questo *geek.*

E quindi ero lì che cercavo disperatamente qualcosa da dire il giorno dopo. (Io i discorsi li preparo solo quando ormai è troppo tardi, il che di solito vuol dire la sera prima della conferenza.) Mi sto scervellando a pensare alla «società della comunicazione», a cosa vuol dire, a ciò che la Nokia e le altre aziende del settore diventeranno in futuro.

E la cosa migliore che mi viene in mente è questa: spiegare il senso della vita.

In realtà non è proprio un «senso. È più una legge della vita, che d'ora in poi sarà chiamata «Legge di Linus». È l'equivalente della seconda legge della termodinamica in fisica, ma anziché spiegare la devoluzione dell'ordine dell'universo riguarda l'evoluzione della vita.

Non sto parlando di «evoluzione» nel senso darwiniano del termine. È un'altra cosa: per il Webrush mi interessava di più il modo in cui evolve la società e il modo in cui noi siamo passati da una società industriale a una società della comunicazione. Cosa succederà dopo? E perché? Volevo che suonasse bene e che fosse una teoria abbastanza sensata da convincere il pubblico per la durata della tavola rotonda. Tutti hanno i propri obiettivi: il mio quel giorno era uscire vivo da una tavola rotonda con due importanti filosofi.

Perché la società evolve? Qual è il propellente? È davvero la tecnologia a trasformare la società? Si tratta di un'idea molto diffusa. È stata veramente l'invenzione del motore a vapore a creare in Europa la società industriale e a portarci poi - tramite la Nokia e i cellulari - alla società della comunicazione? I filosofi sembravano pensarla così. Sembravano interessati ai modi in cui la tecnologia cambia la società.

Io, in quanto tecnico, so che la tecnologia non trasforma un bel niente. È la società a cambiare la tecnologia, e non il contrario. La tecnologia pone solo i limiti di ciò che possiamo fare e di quanto ci costa farlo.

La tecnologia, come gli strumenti che crea, è - almeno per ora - intrinsecamente stupida. É interessante solo per ciò che puoi *farci,* e la forza che spinge tutto è costituita dai bisogni e dagli interessi degli uomini. Oggi non comunichiamo perché abbiamo più mezzi per farlo: lo facciamo perché siamo tutti dei chiacchieroni e abbiamo *voglia* di farlo. Con grande soddisfazione della Nokia.

Così - argomentavo io - per comprendere l'evoluzione della società devi capire cosa motiva veramente le persone. Il denaro? Il successo? Il sesso? Cos'è che spinge le persone a fare ciò che fanno?

L'unico fattore motivazionale che probabilmente nessuno metterebbe in discussione è ovvio: la sopravvivenza. E ciò che definisce la vita, in fondo. La vita non segue ciecamente la seconda legge della termodinamica, ma continua testardamente a *essere* nonostante un universo che sembrerebbe fondamentalmente remare contro la complessità e l'ordine che stanno alla base della vita stessa. E quindi la sopravvivenza è il fattore motivazionale numero uno.

Per stendere una classifica degli altri fattori motivazionali ho dovuto considerare il

modo in cui si ponevano rispetto al puro desiderio di sopravvivere. La domanda non è «Uccideresti per denaro?» ma *«Moriresti* per denaro?» La risposta evidentemente è no. E così possiamo tranquillamente togliere il denaro dall'elenco dei fattori motivazionali fondamentali.

Ma ci sono altre cose per cui le persone sono disposte a morire. Esistono un sacco di storie eroiche di persone (e anche di animali) che decidono di morire per una causa superiore. Quindi la pura sopravvivenza non esaurisce da sola i fattori motivazionali che spingono la nostra società.

Le altre motivazioni che mi vennero in mente per quel discorso a Berkeley erano semplici e durante la tavola rotonda non furono molto contestate. Quindi la maggior parte dei presenti era d'accordo. (Oppure erano semplicemente persone molto educate.) Non ci sono moltissime cose per cui un uomo sia disposto a morire, ma le relazioni sociali sono decisamente fra queste.

Gli esempi di motivazioni sociali che portano qualcuno a dimenticare la priorità della sopravvivenza sono numerosi, dalla letteratura con Romeo e Giulietta (morti non banalmente perché volevano fare del *sesso,* ma perché preferivano morire che perdere la propria relazione sociale) al patriottismo dei soldati disposti a rischiare la propria vita per il paese e la famiglia (ovvero: la società). Quindi possiamo mettere le «relazioni sociali» al secondo posto della nostra classifica.

Il terzo e ultimo fattore motivazionale è il «divertimento. Potrà sembrare triviale, ma si tratta indubbiamente di una forza estremamente potente. Ci sono ogni giorno persone che muoiono facendo qualcosa per puro divertimento. Un esempio? Saltare fuori da un aereo perfettamente funzionante solo per provare un brivido.

E il divertimento non deve essere necessariamente triviale. Può trattarsi di una partita a scacchi, o del divertimento intellettuale di capire come funziona il mondo. Può essere la curiosità e l'esplorazione di un nuovo mondo. Qualsiasi cosa spinga una persona a sedersi dentro un razzo, sopra un godzilliardo di chili di materiale esplosivo solo per poter vedere la terra dallo spazio, può senza dubbio essere definita un «fattore motivazionale».

La sopravvivenza, il tuo posto nell'ordine sociale e il divertimento. Le tre cose che ci fanno fare ciò che facciamo. Tutto il resto è ciò che un sociologo potrebbe chiamare «comportamento emergente;»: schemi di comportamento che emergono dalle regole fondamentali.

Ma non si riduce tutto a un «queste sono le tre cose che motivano le persone;». Se fosse tutto qui, non sarebbe un granché come teoria della vita. Ciò che la rende interessante è che i tre fattori motivazionali hanno un ordine intrinseco, un ordine che compare ovunque vi sia vita. Non siamo solo *noi* a essere motivati da queste tre cose: sono vere anche per forme di vita diverse dagli uomini e costituiscono la progressione naturale di qualsiasi comportamento vitale.

Sopravvivere. Socializzare. Divertirsi. Ecco la progressione. Ed è per questo che abbiamo sottotitolato questo libro *Solo per divertirsi.* Perché tutto ciò che facciamo lo facciamo in ultima analisi per divertirci (almeno se abbiamo avuto la possibilità di progredire fino a questa fase).

Non mi credete?

Pensate a come classifichiamo gli animali in «inferiori» e «superiori». Tutti

sopravvivono. Ma più avanzate sulla scala dell'evoluzione, più è probabile che creiate i primi schemi sociali - anche le formiche, che si trovano abbastanza in basso su quella scala, hanno schemi sociali molto rigidi - che finiscono per progredire fino a giungere al divertimento. Le formiche di solito non giocano molto con il cibo... i gatti sì. E le formiche non si divertono granché nemmeno con il sesso.

Già, prendete qualcosa di elementare (e delizioso) come il sesso. Non sto dicendo che si tratti in sé di uno dei fattori motivazionali fondamentali, ma è un ottimo esempio di un comportamento umano abbastanza fondamentale passato attraverso tutta l'evoluzione della vita. Non c'è dubbio che sia iniziato come puro elemento di sopravvivenza. In fondo anche le piante fanno «sesso nel senso della sopravvivenza, e qualche miliardo di anni fa il sesso era probabilmente una pura questione di sopravvivenza per gli animali monocellulari che si sarebbero lentamente evoluti in geek e altri esseri umani. E non c'è dubbio che il sesso molto tempo fa si sia trasformato da fenomeno di pura sopravvivenza a fenomeno fortemente sociale. Non è solo tra gli umani che si trovano le cerimonie di nozze e tutte le infrastrutture preparatorie a una sana scopata. Pensate alla danza rituale della gru Sandhill, che peraltro si accoppia per la vita. In effetti ogni giorno vengono spese quantità mostruose di energia nei rituali sociali di corteggiamento associati alla semplice questione della riproduzione di tutte le specie.

Divertimento? Anche quello, ve lo assicuro. Non solo tra gli umani, ma probabilmente non è una coincidenza il fatto che la maggior parte delle specie evolute del pianeta sembrino essere anche quelle che apprezzano particolarmente gli aspetti ludici del sesso.

La progressione dalla sopravvivenza al comportamento sociale al divertimento si trova dappertutto. Pensate alla guerra: decisamente una questione di sopravvivenza quando l'unico modo per arrivare a una fonte d'acqua era uccidere chi la voleva tenere per sé. In seguito la guerra divenne uno strumento finalizzato al mantenimento dell'ordine sociale. E con l'avvento della CNN si è trasformata in intrattenimento. Che ci piaccia o no, sembra che si tratti di una progressione inevitabile.

La civiltà stessa segue questo schema generale. In origine si trattava di un modo per assicurare la sopravvivenza tramite la cooperazione e il potere del numero. Non è una caratteristica esclusiva degli umani. La maggior parte degli animali e anche le piante creano società finalizzate a una migliore sopravvivenza tramite l'aiuto reciproco. È molto interessante come la società stessa passi da un fondamento legato alla sopravvivenza a uno più sociale: tutte le civiltà umane hanno finito per costruire strade e canali di comunicazione migliori per poter socializzare meglio.

E alla fine anche la civiltà diviene una questione di divertimento. Pensate all'Impero Romano, famoso non solo per le strade che costruiva e per la solidità dell'ordine sociale, ma - soprattutto in un periodo tardo - per i suoi divertimenti.

Oppure pensate agli Stati Uniti odierni. Qualcuno ha dei dubbi sul fatto che l'industria cinematografica e quella dei videogiochi ci stiano accompagnando verso una società del divertimento? Non molto tempo fa si trattava di mercati di nicchia, oggi sono tra i settori più importanti nell'economia del paese più ricco del mondo.

E la cosa che trovo interessante in quanto tecnico è il modo in cui questo schema si ripete nelle tecnologie che creiamo. Il periodo iniziale delle tecnologie moderne è

stato chiamato Età Industriale, ma in realtà si dovrebbe chiamare Età della Sopravvivenza Tecnologica. La tecnologia fino a non molto tempo fa serviva quasi esclusivamente per sopravvivere meglio: tessere meglio i tessuti, spostare le merci più velocemente. Fu questo a scatenare tutto ciò che seguì.

Il periodo attuale viene denominato Età dell'Informazione. È un grande passaggio. La tecnologia viene usata per la comunicazione e per la diffusione delle informazioni - un comportamento estremamente sociale - e non solo per sopravvivere meglio. Internet (e il fatto che tanta parte della nostra tecnologia stia iniziando a muoversi nella medesima direzione) è un grande cartello stradale dei nostri tempi: significa che le persone nei paesi industrializzati stanno iniziando a dare per scontata la questione della sopravvivenza e all'improvviso la seconda fase delle tecnologie - il loro aspetto sociale, quello che implica il loro uso non solo per vivere meglio ma come parte integrante della vita sociale - diviene assai eccitante.

Incombe anche, naturalmente, l'obiettivo finale. Dopo la società dell'informazione, la società del divertimento. Un luogo in cui internet e la comunicazione wireless 24 ore su 24 vengono dati per scontati e non finiscono più sulle prime pagine dei giornali. Un'epoca in cui la Cisco sarà il vecchio mercato e la Disney Corporation sarà la padrona del mondo. Un'epoca probabilmente non troppo distante nel futuro.

Cosa significa tutto ciò? Probabilmente non molto. In fondo la mia teoria sul senso della vita non è una *guida* su ciò che si dovrebbe fare. Al massimo dice: «Sì, puoi anche contrapporti a questa cosa, ma in ogni caso l'obiettivo finale della vita è divertirsi.»

In certa misura questo spiega perché la gente provi il desiderio di lavorare via internet su progetti come Linux. Per me e per molte altre persone Linux è stato un modo per depennare dall'elenco due motivazioni in una volta sola. Dando per scontata la sopravvivenza, Linux ci ha dato sia il divertimento di una sfida intellettuale sia le motivazioni sociali associate all'essere parte della sua creazione. Forse non ci siamo visti molto di persona, ma le nostre e-mail non erano dei semplici scambi di informazioni. Con la posta elettronica si possono creare legami d'amicizia e altri rapporti sociali.

Questo probabilmente significa anche che - se e quando incontreremo un'altra forma di vita intelligente nell'universo - le prime parole che le sentiremo dire non saranno «Portatemi dal vostro capo. É più probabile che dica qualcosa del tipo: «Facciamo festa, amici!»

Ma naturalmente mi potrei sbagliare.

Glossario

Nota: per informazioni di carattere generale su Linux rimandiamo ai siti http://www.linux.org (sito ufficiale) e http://www.linuxvalley.com (in italiano).

ASSEMBLER: programma che traduce in → linguaggio macchina un programma scritto in linguaggio → assembly. È l'opposto di un → disassembler.

ASSEMBLY: linguaggio di programmazione di basso livello. Strutturalmente simile al → linguaggio macchina, utilizza nomi convenzionali invece di codici istruzioni ed etichette simboliche invece di locazioni di memoria.

BACO: errore di programmazione di un programma informatico che ne può causare un funzionamento errato. Molti diffuso anche in Italia il termine inglese bug.

BASIC: acronimo di Beginner's All-purpose Symbolic Instruction Code, linguaggio di programmazione nato nel 1964. Ne sono state realizzate molte versioni differenti.

BETA: versione di un programma informatico precedente alla diffusione ufficiale, distribuita ai → beta tester perché ne provino la funzionalità.

BETA TESTER: utenti di prova della versione → beta di un programma informatico.

BIOS: acronimo di Basic Input/Output System, è l'interfaccia tra l'hardware e il → kernel del sistema operativo. BUG: → baco.

BUNDLE: software distribuito gratuitamente in associazione con un componente hardware o un altro software (Es: un programma di grafica offerto in bundle insieme a una macchina fotografica digitale).

C: linguaggio di programmazione nato nel 1972 per tradurre il sistema operativo → UNIX e renderlo portabile su un alto numero di elaboratori.

CHIP: componente di materiale semiconduttore su cui vengono miniaturizzati circuiti integrati con diverse funzionalità.

CLIENT: dispositivo o programma che - all'interno di una rete -viene utilizzato da un utente per contattare una sorgente di informazioni situata in un altro punto della rete, il → server.

CLONE: elemento hardware (spesso un intero computer) basato sulle specifiche di un originale, al quale può essere più o meno identico.

CODICE: in informatica, sistema di simboli e regole utilizzato per rappresentare delle istruzioni o informazioni in un linguaggio comprensibile da un computer.

CODICE SORGENTE: → codice scritto in un linguaggio di alto livello. Prima dell'esecuzione deve essere compilato con un → compilatore.

COMPILATORE: programma informatico che traduce un linguaggio di alto livello (→ codice sorgente) in → linguaggio macchina eseguibile.

CONTROLLER: dispositivo in grado di regolare un flusso di dati, inviare comandi ad altri dispositivi ed eseguire una interrogazione per stabilire quale di questi ha richiesto un servizio.

CRACKARE: in gergo informatico, superare i dispositivi di sicurezza di un programma, di una rete, di un computer o di un dispositivo informatico in generale.

CREDIT LIST: elenco dei responsabili di un programma informatico, generalmente allegato ai programmi → open source sotto forma di file.

DEBUG: fase successiva alla programmazione, consiste nell'eliminare i → bachi da un programma.

DISASSEMBLER: programma che traduce in linguaggio → assembly un programma scritto in → linguaggio macchina. É l'opposto di un → assembler.

DRIVE: dispositivo per la lettura e scrittura di dati su un disco magnetico o ottico, gestito da un → controller.

DRIVER: programma che consente a un computer di comunicare con una periferica, come una stampante o un lettore CD.

EMULAZIONE DI TERMINALE: programma che trasforma un personal computer in un dispositivo che simula le funzioni di terminale remoto di un computer a cui si collega.

FREE SOFTWARE: traducibile in italiano con «software libero (e non necessariamente con «software gratuito), l'espressione si riferisce alla libertà dell'utente di eseguire, copiare, distribuire, studiare, cambiare e migliorare il software. L'accesso al → *codice sorgente* ne è un prerequisito. Più precisamente, esso si riferisce a quattro tipi di libertà per gli utenti del software: a) libertà di eseguire il programma, per qualsiasi scopo; b) libertà di studiare come funziona il

programma, e adattarlo alle proprie necessità; c) libertà di ridistribuire copie in modo da aiutare il prossimo; d) libertà di migliorare il programma e distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunità ne tragga beneficio. Si veda il sito http://fsfeurope.org.

FTP: acronimo di File Transfer Protocol, protocollo internet per lo scambio di file tra computer collegati alla rete.

GEEK: in gergo informatico è un incrocio tra il «secchione» e lo «smanettone»; per esteso chiunque sia molto appassionato di informatica.

GNU: progetto sostenuto dalla Free Software Foundation per creare un ambiente di sviluppo assolutamente indipendente dall'architettura hardware/software. Realizzato su base volontaria, GNU viene ceduto gratuitamente e con una licenza che consente a chiunque il diritto di apportare qualsiasi modifica (→ *open source).* Si veda il sito http://www.gnu.org/home.it.html.

GPL: acronimo di General Public License, è un tipo di licenza per software che rovescia i termini del copyright classico proponendo il software come libero e modificabile. Ne deriva il verbo gergale *GPLlare* (si legga «gipiellare», ovvero distribuire un prodotto con una licenza di tipo GPL.

HACKER: in gergo informatico l'*hacker* è un programmatore molto abile. Linus Torvalds lo usa sempre in questo senso. In seguito il termine è passato a designare chi è in grado di accedere illegalmente a sistemi informatici protetti.

HISTORY FILE: file che riassume la storia delle versioni successive di un programma e cita i nomi dei suoi responsabili. Generalmente allegato ai programmi → *open source.*

KERNEL: lo strato più interno di un sistema operativo, con procedure generalmente invisibili all'utente se non nei sistemi operativi → *open source.*

LINGUAGGIO ASSEMBLY: → *Assembly.*

LINGUAGGIO BASIC: → *BASIC.*

LINGUAGGIO C: → *C.*

LINGUAGGIO MACCHINA: linguaggio di programmazione di basso livello, con operazioni codificate da sequenze di cifre binarie e direttamente interpretabili dal processore.

LISP: acronimo di List Processor («elaboratore di listo). Linguaggio di programmazione per la manipolazione di simboli creato nel 1958 al MIT.

LOGIN: sessione di un singolo utente all'interno di una rete o di un computer multiutente. Per esteso il termine viene utilizzato per indicare il nome dell'utente e della sessione.

MAINFRAME: computer multiutente generalmente utilizzati come controllori di reti e di sistemi o come gestori di banche dati.

MP3: formato di compressione per file audio, permettere per esempio di comprimere un'intera canzone in un file di pochi megabyte e la rende quindi facilmente trasmissibile via internet.

MULTITASKING: capacità di un computer di svolgere più funzioni, ovvero elaborare più programmi, contemporaneamente.

MULTITHREAD: metodo per l'elaborazione di processi multipli e separati all'interno della stessa applicazione.

NEWSGROUP: conferenza elettronica (in ambiente usenet - internet) dedicata alla discussione di uno specifico argomento.

OPEN SOURCE: sistema di sviluppo basato sul principio che chiunque possa apportare modifiche al → *codice sorgente* di un programma. Si veda il sito http://www.opensource.org.

PAGING: processo per realizzare un sistema di memoria virtuale in un computer trasferendo pagine di informazione tra memoria centrale e memorie ausiliarie.

PARTIZIONE: suddivisione del disco fisso di un computer in blocchi indipendenti.

PATCH: aggiornamento di un software realizzato allo scopo di eliminare uno o più difetti presenti in una → release o di modificarne alcune funzionalità.

PIXEL: acronimo deformato di PICture Element. È l'elemento più piccolo per la formazione di schermate grafiche su uno schermo. La definizione grafica viene generalmente misurata in base al numero di pixel (o dot) presenti in un pollice quadrato di schermo.

PROMPT: messaggio inviato da un computer a una periferica (generalmente a video) per informare l'utente che il sistema è pronto a ricevere un comando. In DOS il prompt si presenta così: C:\>

RAM: acronimo di Random Access Memory, è la memoria centrale volatile di un computer, utilizzata per l'esecuzione di programmi.

RELEASE: versione di un programma. Le release sono numerate in modi differenti, ma il metodo più frequente consiste in cifre composte da serie di numeri crescenti divisi da punti: i numeri più a destra indicheranno modifiche via via meno sostanziali di quelli a sinistra. Esempio: Software 3.5.1

ROM: acronimo di Read Only Memory. Memoria non volatile di sola lettura, memorizza in modo permanente le procedure necessarie al funzionamento del computer.

SERVER: computer che fornisce servizi a un altro computer detto → client o agli utenti della rete locale.

SHELL: parte del sistema operativo con funzioni di interfaccia tra un utente e i programmi del sistema, il cui uso risulta così facilitato.

SLOT: presa o alloggio in un cui inserire un connettore, una scheda o più in generale un elemento predisposto.

SMP: linguaggio di programmazione nato al California Institute of Technology all'inizio degli anni Ottanta.

SOCKET DI DOMAIN: combinazione di un indirizzo IP con un indirizzo di porta. Viene utilizzato in internet per collegarsi a un'applicazione specifica (indirizzo di porta) su un computer specifico (indirizzo IP).

UNIX: sistema operativo multiutente nato nel 1969 nei Bell Labs per opera di Ken Thompson. In origine era scritto in linguaggio → *assembly,* ma è stato in seguito tradotto in → *C* per essere utilizzato su un'ampia gamma di elaboratori. È alla base dello sviluppo di Linux.

VIRGOLA MOBILE: tipo di calcolo complesso per elaborazioni grafiche, soprattutto in 3D.

WIRELESS: letteralmente «senza fili». Indica un elemento o un sistema informatico in grado di collegarsi in rete senza l'utilizzo di una connessione fisica. Esempio: la funzione WAP dei telefoni cellulari.

Ringraziamenti

Gli autori desiderano ringraziare le seguenti strutture per il ruolo sostenuto nella creazione di questo libro... o perlomeno per averne reso la creazione più divertente. (Nessuno di loro ci ha pagato un dollaro. Il che è un gran peccato.)

FM 107.7 the Bone. Classic Rock That Rocks; Zelda's Restaurant, Capitola; Kiva Retreat House, Santa Cruz; Haga-shi West Restaurant, Palo Alto; Malibu Gran Prix, Redwood Shores; Bodega Bay Lodge, Bodega Bay; Saturn Café, Santa Cruz; Café Marmalade, Ross; Half Moon Bay Boardshop, Half Moon Bay; Santa Cruz Billiards, Santa Cruz; Café Reyes, Point Reyes Station; California Sushi and grill, San José; Santa Clara Golf and Tennis Club, Santa Clara; Ideal Bar and Grill, Santa Cruz; Silver Peso Bar («Dove-Ha-Suonato-Janis»), Larkspur; Ro-sie McCann's Iri-sh Pub and Restaurant, Santa Cruz; Mayflower Inn, San Rafael; Grover Hot Springs State Park, Markleeville; Left Bank Restaurant, Larkspur; Potrero Brewing Company, San Francisco; The Rice Table, San Rafael; Ross Valley Swim and Tennis Club, Kentfield; Fal-len Leaf Lake Marina, Fallen Leaf Lake; Peet's Coffee and Tea, Greenbrae; Hawthorne Lane Restaurant, San Francisco; Indian Springs Resort, Calistoga; Samurai Sushi, Sausalito; Blow-fish Sushi, San Francisco; Paramount's Great America, Santa Clara; Robata Grill Sushi, Mill Valley; Buckeye Roadhouse, Mill Valley; Barnes and Noble, San José; Sushi Ran, Sausalito; 23 Ross Common, Ross; KFOG-104.5 FM; Rutherford Grill, Rutherford; In-N-Out Burger, Santa Rosa; Seto Sushi, Sunnyvale.